

Tietokonekulttuurin erikoislehti

Fraktaaliavaruuksia tutkimaan

Mandelbulb
Mandelbox
Bulbbox
Mandelbrot

Taistelu internetistä:
vapauttajat vs rajoittajat

Klassikkokoneet
nykymaailmassa

C-64, Spectrum, Amiga, PC

Thinkpadin
suojauskset auki

Rakenna oma
saunatonnttu



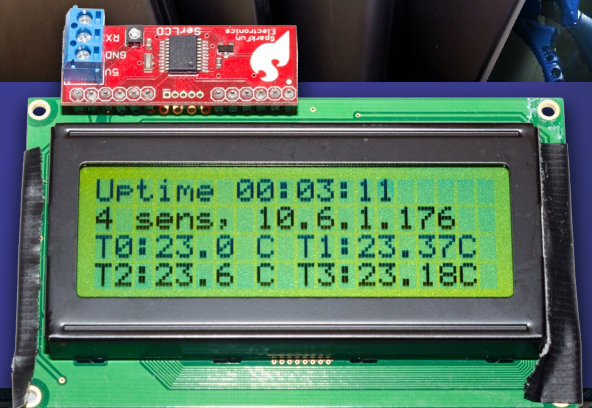
Neuvostoliiton kotimikrot ja ohjuslaukaisimet



Tee itse QML-käyttöliittymiä



Nojatuolikeisarin klassikot avoimina:
FreeCiv, FreeOrion, OpenTTD



3 Pääkirjoitus

4 Ilmaiset strategiaklassikot

Avoimen lähdekoodin versiot varmistavat näiden pelien kuolemattomuuden.

8 Taistelu internetistä: vapauttajat vs rajoittajat

Internetin kehityksen suuntaan vaikuttavat hyvin monenlaiset toimijat.

12 Tuomiopäivän koneet

Ydinsota yhden päätöksen päässä.

15 Ei näin! Atari Jaguar, kissapedon kuolinkorinat

Jaguar ei ollut aivan sitä, miltä se näytti.

16 Vanhojen bittien pelastajat

Arkistoinnilla varmistetaan, ettei menneiden vuosikymmenten digitaalinen kulttuuri katoa jäljettämiin.

20 3D-rajapintojen lyhyt historia

Ennen kuin syvennymme OpenGL-ohjelmointiin on hyvä kerrata vähän historiaa.

22 Mitä klassikkokoneille kuuluu nykyään?

Jotkin tietokonemallit eivät kuole pois sitten millään.

28 QML — deklarativisen käyttöliittymän alkeet

Näyttäviä käyttöliittymiä ilman kovinkaan kummoista ohjelmointitaitoa.

33 Viestejä ylihuomisen lukijalle

34 Mandelbrotin 3D-perilliset

Marraskuussa 2007 päästiin viimein aidon 3D-Mandelbrotin jäljille.

40 Neuvostoliiton ja itäblokin tietokoneiden historia

Itätietotekniikan rujot mutta innovatiiviset laitteet ihastuttavat tänäkin päivänä.

45 Bytebeat-musiikkia

Pienet koodinpätkät tuottavat yllättävän musiikkimaista ääntä.

46 Arduinosta saunavahti

Lopeta saunalla ravaaminen — kotitonttu kertoo lämpötilan.

48 Päihitä Thinkpadin suojat

Kuinka PC-tietokoneita suojataan, ja kuinka lukkoon jäänyt Thinkpad palautetaan käyttökuuntoon?

52 Seuraavan Skrollin aiheita



Ville-Matias Heikkilä
päätoimittaja

Skrolli

Tietokonekulttuurin erikoislehti

Yhteydenotot toimitus@skrolli.fi
ircnet — #skrolli

Päätoimittaja Ville-Matias Heikkilä

Muu toimitus Tapio Berschewsky, Mikko Heinonen,
Ronja Koistinen, Anssi Kolehmainen,
Sade Kondelin, Ninnu Koskenalho,
Toni Kuokkanen, Juho Lehtinen, Elias
Linjama, Risto Mäki-Petäys

Avustajat Matti Hämäläinen, Jari Jaanto, Risto
Järvinen, Eetu Korhonen, Jarkko Lehti,
Ville Ranki, Kalle Viiri, Mikko Rasa, Aki
Sivula

Julkaisija Alternative Party ry

Painopaikka Tammerprint, Tampere,
ISSN 2323-8992 (painettu)
ISSN 2323-900X (verkkójulkaisu)

Kokonaiskuvaa ja jatkuvuutta

Mitä järkeä tässä on? On 2010-luku ja jotkut perustavat painetun tietotekniikkalehden, vaikka netti on paljon nopeampi! Täyttävät vielä sen jollain ohjelmoinnilla, rakentelulla ja historiallisella päätöksellä. Eihän tuollainen kiinnosta ketään! Missä talousuutiset ja tuotevertailut?

Skrolli-lehti on syntynyt maailmaan, jossa tietokoneet ovat etupäässä kulutushyödykkeitä. Kaikki käyttävät niitä, mutta harvempaa kiinnostaa ymmärtää niitä. Jopa suomalaiset tietokonelehdet ovat nykyään lähinnä osto-oppaita, joista inhimillinen tekeminen ja innostus on sysätty pois kulutuskeskeisyyden tieltä. Aikoinaan nämä lehdet inspiroivat kokonaista uutta osaaajasukupolvea, mutta nykyään eivät osaaajiksi haluavat löydä niistä enää mitään kiinnostavaa.

Eihän tietokonelehtiä nykyään sinänsä tarvitsisi, koska onhan netti täynnä luovaa tietokonekulttuuria, jossa osaamista arvostetaan edelleen. Tarjolla on tuhansittain mahdollisia viiteryhmiä ja erikoistumislokeroita. Nettiyhteisöjen ongelmana on kuitenkin pirstaleisuus: eri suuntiin kehittyneet tietokonekulttuurin rönsyt eivät aina ymmärrä toisiaan tai edes tiedä toisistaan. Kulttuuri tarvitsee tämän vuoksi jonkun, joka rakentaa siltoja kuppikuntien välille ja antaa monipuolisen kuvan erilaisista tekemismuodoista suhteessa toisiinsa. Tämä ei onnistu pelkillä tykkäyslaskureilla, vaan tarvitaan ihan oikeaa toimitustyötä.

Pirstaleisuuden lisäksi tietokonekulttuuria vaivaa katoavaisuus. Jos jokin on olemassa pelkkinä bitteinä, jonkun on erikseen nähtävä vaivaa arkistointiin. Yksittäisen nettisivuston lopettaessa saattaa hävitä suuri osa kokonaisesta alakulttuurista, mikä usein huomataan vasta kantapään kautta. Tästä syystä on hyvä pitää historiallista viitekehystä esillä silläkin uhalla, että sitä pidetään ”retroiluna”.

Skrollin tarkoitus on olla tietokonekulttuurin yhteinen lehti, jolle sekä kokonaiskuva että jatkuvuus ovat tärkeitä asioita. Se painetaan paperille, joka kestää aikaa paremmin kuin levyt ja muistipalikat. Se haluaa tuoda esiin kaikenlaisia tietoteknisiä tekemiskulttuureja jäämättä tiettyjen kuppikuntien äänitorveksi. Se yrittää tarjota tietotekniisiin asioihin laajempaa ja kestävämpää näkökulmaa kuin muut.

Jatkuvuusteema näkyy vahvasti Skrollin ykkösnumerossa. Kunnianosoituksena vanhoja tietokonelehtiä kohtaan vedämme monista niiden aiheista jatkumoa nykyaikaan: Mitä 80-luvun mikrotietokoneilla tehdään nykyään? Ketkä arkistivat vanhaa tietokonekulttuuria? Millaisia fraktaaleja on löytynyt Mandelbrotbuumin jälkeen? Millaisia ovat takavuosisikymmenten suosikkipelien uusversiot?

Näin vankan historialliselta pohjalta Skrollin on hyvä ponnistaa tulevaisuuteen. Lehdellämme on kuitenkin vielä paljon tekemistä ennen kuin sen toiminta vakiintuu, ja kaipaamme tässä myös lukijoiden apua. Levittäkää sanaa, tarjotkaa sisältöä, varmistaakaa tietokonekulttuurin lipunkantajan jatkuvuus!



Ilmaiset strategiaklassikot

Mitä pitäisi ajatella, kun 20 vuotta vanhasta pelistä ei pysty irrottamaan käsiään tai mieltään? Onko kyseessä nostalginen obsessio vai olivatko pelit 90-luvulla oikeasti niin hyviä? Näiden klassikoiden ilmaisversioilla voit tarkastaa, onko naavaparran rakkaustarinassa idun itua. Teksti: Kalle Viiri Kuvat: Kalle Viiri, Ville-Matias Heikkilä

Monet pelit päätyvät kiihkeään fanituksen kohteiksi. Civilization, Colonization, Transport Tycoon ja Master of Orion kilpailevat kuitenkin kokonaan omassa sarjassaan. Näitä joka pelaajan keisaripelejä on pelattu tauotta ysärliltä saakka, eikä loppua näy. Etenkään kun nykyään on tarjolla laadukkaita avoimia ja ilmaisia versioita alkuperäisklassikoista.

Skrolli testasi pelien uusversiot ajan tuomalla näkemyksellä. Kokemusta verrattiin alkuperäiseen ja mietittiin, vieläkö rakkaus kantaa. Peliklassikot osoittautuivat rosoisesta ulokuo-
resta ja lievistä bugisuudestaan huolimatta erinomaiseksi viih-
teeksi. Nämä teokset eivät kuole.

FreeCiv vs Sid Meier's Civilization

Sid Meier's Civilization on strategiapelitaivaan kiintotähti, joka jatko-osineen on dominoinut myyntilistoja ja faniensa ajankäyt-
töä jo yli kahden vuosikymmenen ajan.

Sarjan peleissä luotsataan omaa historiallista kansakun-
taa kolkolta kivilaudelta aina lähitulevaisuuteen asti. Muita
imperiumeja vastaan kilpaillaan raa'an sotilasmahdin lisäksi
kulttuurilla, rahalla, teknologisella kehityksellä ja ovelalla dip-
lomatialla.

Pelin voittaminen tapahtuu ensimmäisessä osassa varusta-
malla avaruusalus Alfa Kentauri -tähdelle. Myöhemmissä osissa
sivilisaationsa saa kruunattua voittajaksi myös kulttuurin,
talouden tai diplomatian keinoin. Voitto tipahtaa perinteisesti
myös nuijimalla kaikki muut kansakunnat maan rakoon.

FreeCiv

FreeCiv on C-kielellä toteutettu avoin Civilizationin kaltainen
peli, joka tuntuu erityisen tutulta ensimmäistä kahta Civiliza-
tionia pelanneelle. Pelin tarjoamat yksiköt, rakennukset ja tek-
nologiat on lainattu pääosin suoraan esikuvilta. Myös grafiikat
tuovat vahvasti mieleen Civilization II:n. Tänä päivänä ulkoasua

ei voi pitää kauniina, mutta se on erittäin funktionaalinen.

Verkon yli pelaaminen on mahdollistanut aktiivisen pelaaja-
yhteisön sisäiset voimainmittelöt. Tekoälyvastustajatkin pelaa-
vat kohtuullisen hyvin, joten pelistä on iloa myös yksinpelaa-
jille. Moninpeli ja mielenkiintoisemmat yksinpelikokemukset
ovatkin parasta mitä FreeCivillä on tarjota.

Alkuperäinen Civilization-sarja on tarjonnut aina muokat-
tavia ominaisuuksia pelaajilleen, eikä FreeCiv jää tällä saralla
jälkeen esikuvastaan. Suurta osaa pelin toiminnoista ohjaavat
helposti muokattavat sääntöpaketit. Paketteja muokkaamalla
voi muun muassa lisätä ja poistaa yksiköitä, teknologioita ja
kansakuntia peliinsä. Paketteihin voi lisätä myös erikoistapah-
tumia Lua-skriptien muodossa.

FreeCol vs Sid Meier's Colonization

Sid Meierin luoman suosittu Civilization-sarjan spinoff-peli
Colonization asettaa pelaajan siirtomaaherran saappaisiin
1400-luvun lopussa. Pelaaja hallinnoi alussa yhtä laivaa ja
pientä joukkoa siirtolaisia, ja tavoitteena on luoda taloudelli-
sesti elinvoimainen siirtokuntaverkosto uudelle mantereelle.

Ahkeralla tuotannolla ja kaupankäynnillä siirtokunnat vau-
rastuvat ja niistä kehittyy hiljalleen omavaraisempia. Haastetta
tuovat verorahoja riistävä kuningas, toiset lääniniä havittelevat
siirtomaavallat ja herkkänahkaiset intiaanit.

Loppuhuipennuksena pelaaja pääsee puolustamaan siirto-
kuntiaan vanhan emämaansa valtavaa punatakkiaarmeijaa vas-
taan itsenäisyys sodan merkeissä — vapaus tai kuolema!

FreeCol

Java-kielellä toteutettu FreeCol pyrkii mallintamaan alkuperäi-
sen Colonizationin toimintaa mahdollisimman tarkasti. Ulko-
asultaan ja käyttöliittymältään FreeColia on kuitenkin päivi-
tetty reippaasti. Peliin on lisätty verkkomoninpeli ja runsaasti
muokattavuutta.



FreeCol



FreeCiv

Vaikka lähtökohtaisesti onkin pyritty samanlaiseen pelikokemukseen kuin alkuperäisessä Colonizationissa, on FreeColiin kertynyt paljon uuttakin. Colonizationissa rehottavat bugit ja laitteistotehoja säästävät rajoitteet ja yksinkertaistukset voi FreeColissa säätää päälle tai pois makunsa mukaan. Uusia siirtomaavaltoja on lisätty: siirtokuntia rapakon taakse rakentelevat tuttujen Alankomaiden, Englannin, Espanjan ja Ranskan lisäksi myös Portugali, Ruotsi, Tanska ja Venäjä. FreeCol on virallisesti alpha-vaiheessa. Vaikka pelin kotisivuilla varoitellaan sen vilisevän vielä bugeja, ei niitä testauksen

aikana juuri tullut vastaan. Bugeja isompi harmi pelaajalle ovat kömpelö ja ajoittain epäintuitiivinen käyttöliittymä ja ulkoasu, joka paremmasta resoluutiostaan huolimatta on karun oloinen värikkään Colonizationin rinnalla. Odotettua heikommin pelissä toimii myös tekoäly, joka tasapainottaa huonoa pelaamistaan huijaamalla rankasti oletusasetuksilla.

Kaiken kaikkiaan FreeCol on viimeistelemättömän ja hieinan keskeneräisen oloinen mutta toimiva ja kehittyvä modernisointi Colonization-klassikosta. Voikin olettaa, että tämänhetkiset ongelmat katoavat vähitellen kehitystyön edetessä.

Myös kaupallista tarjolla

Transport Tycoon ja Master of Orion on sarjoina kauan sitten kuopattu, mutta Civilization porskuttaa yhä. Viides osa peliin julkaistiin syksyllä 2010. Sarjan uusin versio on tosin jo varsin kaukana alkuperäisestä Civilizationista. Monet sarjaa seuranneet pelaajat pitävätkin neljättä Civiä parhaana liittona uusien mekaniikkojen ja vanhojen perusperiaatteiden välillä. Kummatkin pelit saa halutessaan Steamista.

Myös Colonization on saanut kaupallisen modernisoinnin. Firaxis rakensi Civilization IV:n moottorin päälle uuden Colonization-versioinnin kolme vuotta sitten. Tätä kritisoiinkin maksulliseksi modiksi, mutta peli itsessään jaksoi viihdyttää yhä. Myös Civilization IV: Colonization löytyy Steamista.

Alkuperäisten pelien metsästäminen on hieman hankalampaa. Gog.comista löytyy nykytietokoneiden kanssa yhteensopiviksi saatetut versiot alkuperäisestä Sid Meier's Colonizationista ja Master of Orionista. Muita pelejä joutuu etsimään netti-huutokauppojen kautta.



Simutrans

OpenTTD



Simutrans ja OpenTTD vs Transport Tycoon

Transport Tycoon on lukuisia jatko-osia poikunut rakentelupeli, jossa pelaajan tehtävänä on luoda toimiva ja tuottoisa kuljetus-yritys.

Rahtia ja matkustajia siirrellään kaupunkien ja teollisuuslaitosten välillä erilaisilla kulkuneuvoilla autoista ja höyryjunista suihkukoneisiin. Nopealla kuljetuksella maksimoidaan yrityksen tuottama voitto, joka käytetään kuljetuslinjojen optimaaliksi hiomiseen.

Erilaisia kulkuvälineitä reitteineen on tarjolla mukava valikoima, josta saa kasattua helposti haluamansa kaltaisen pie-noisrautatien korvikkeen.

Simutrans ja OpenTTD

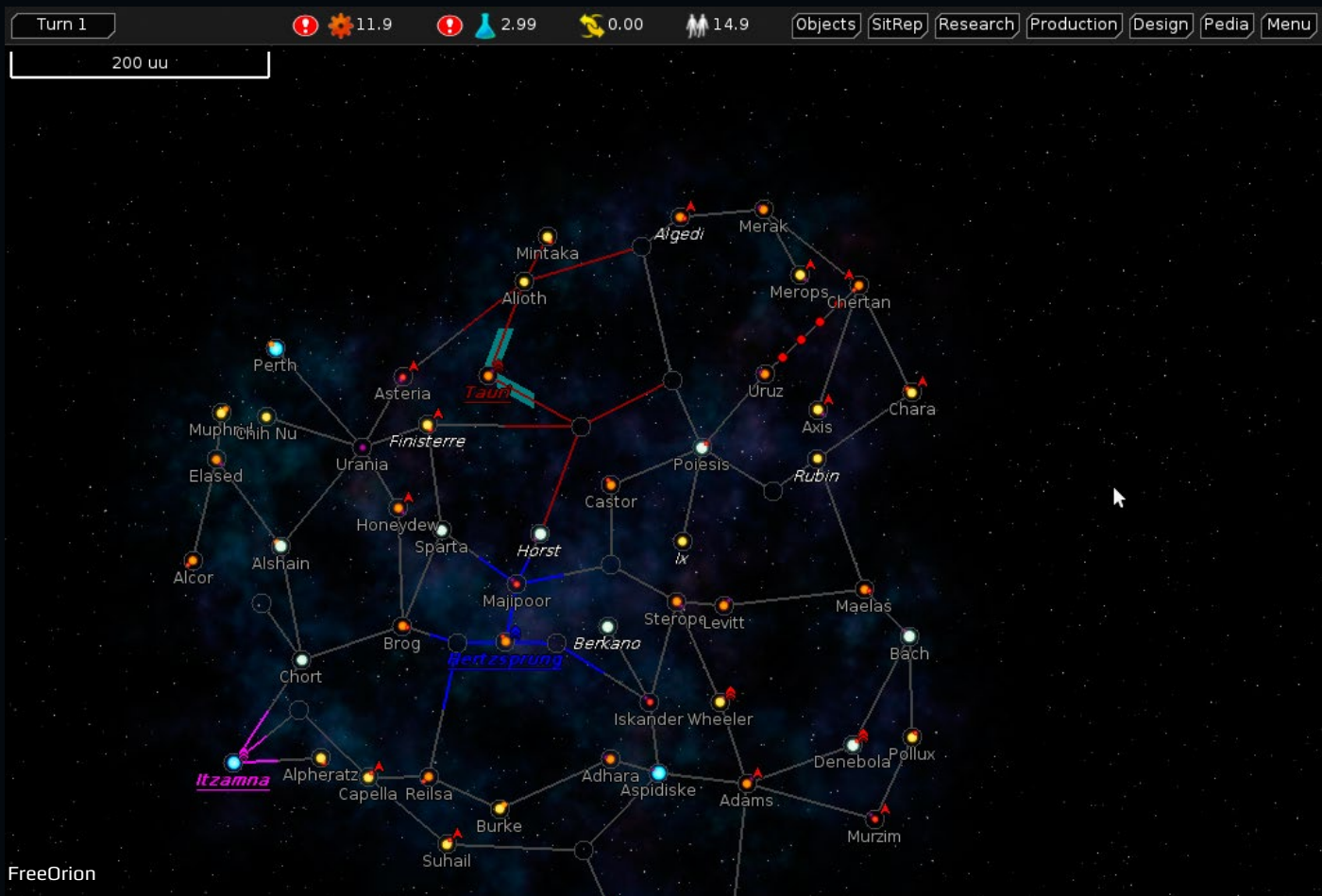
Transport Tycoonin konsepti on poikunut kaksi merkittävää avointa jäljitelmää, OpenTTD ja Simutrans. OpenTTD pyrki jäljittelemään alkuperäisen pelin tunnelmaa mahdollisimman tarkasti. Simutrans puolestaan irtautuu hieman Transport Tycoonin pelimekaniikoista luoden erilaisen pelikokemuksen.

Ensi vilkaisulla Simutrans ja OpenTTD eivät eroa toisistaan juurikaan. Lyhyen pelaamisen jälkeen yksi ero käy kuitenkin tuskallisen selväksi: toimivan kuljetusverkoston rakentaminen on Simutransissa huomattavasti tarkempaa kuin OpenTTD:ssä,

jonka tapa mallintaa rahtiliikennettä on lähempänä Transport Tycoonin naiivia simulaatiota.

Kuljetuslinjojen rakentaminen toisiaan tukeviksi on Simutransissa selvästi OpenTTD:tä tärkeämpää. Pelaajan yritys jää reilusti tappiolliseksi, jos suunnitellut linjat eivät toimi saumattomasti yhdessä. Ratkaisu on realistinen ja lisää peliin haastetta, mutta toisaalta OpenTTD:n rennompi tapa käsitellä matkustaja- ja rahtiliikennettä antaa pelaajalle vapaammat kädet luoda unelmiensa liikenneverkko huolehtimatta paljoa kannattavuudesta.

Sekä Simutrans että OpenTTD ovat visuaaliselta ilmeeltään alkuperäisen Transport Tycoonin kaltaisia ja sopivat hyvin nykyaikaisillekin ruuduille. Käyttöliittymä ja grafiikat ovat toimivia, mutta molemmissa on kömpelötkin puolensa. Esimerkiksi pitkien linjojen muokkaaminen Simutransissa vaatii enemmän naksuttelua kuin luulisi tarpeelliseksi, samoin pysäkkien ja teiden rakentelu OpenTTD:ssä. Simutransin englanninkielinen paketti sisältää myös pieniä keskeneräisyyksiä: osa rakennusten ja ajoneuvojen nimistä on jäänyt laiskasti saksankielisiksi. Sekä Simutrans että OpenTTD tukevat pelin laajaa muokkaamista moditiedostojen avulla. Molemmissa on myös mahdollisuus verkkomonipeliin.



FreeOrion vs Master of Orion

MicroProsen Master of Orion kuuluu avaruusstrategiapelien kärkinimistöön. Siinä pelaaja ottaa luotsattavakseen avaruutta kansoittamaan lähtevän sivilisaation ja pyrkii kasvattamaan sen suureksi imperiumiksi naapurilajien hirtäessä vastaan.

Naapureiden kanssa voi usein toimia diplomaattisesti esimerkiksi kauppaa käymällä tai rauhansopimuksia tekemällä. Konfliktitilanteet ratkaistaan vuoropohjaisilla taisteluilla.

Kehittämällä uusia teknologioita pelaaja saa aluksiinsa uusia osia, jotka voivat vaikuttaa pelin kulkuun ratkaisevasti. Uudempi alus voi täryttää tuhaksi kokonaisen laivueen vanhentuneita aluksia.

Pelin voittaa lopulta pelaaja, joka saa tarpeeksi ääniä määräajoin järjestettävässä äänestyksessä. Sotaisamman pelaajan on mahdollista valloittaa tarpeeksi ääniä itselleen voittoa varten tai pyyhkäistä kaikki kansat pois tieltä ja unohtaa vaalit kokonaan.

FreeOrion

FreeOrion ei pyri olemaan identtinen minkään Master of Orion -sarjan pelin kanssa vaan pikemminkin itsenäinen jatko-osa sarjalle. Pelimekaniikoiltaan FreeOrion yhdistää Master of Orion -sarjan pelien ominaisuuksia ja tuo mukaan vähän uuttakin.

Yksi alkuperäiseen Master of Orioniin verraten silmiinpistävimmistä muutoksista on tähtien väliin merkityt kulkulinjat, jotka ovat ainoita mahdollisia reittejä tähtienväliseen

liikenteeseen. Sarjan ensimmäiseen osaan totuneelle tulevat uutena myös useita planeettoja sisältävät aurinkokunnat ja lineaarisempi teknologiaapu. Uusi resurssijärjestelmä myös erottaa teknologian kehityksen teollisuudesta aiempaa selkeämmin.

FreeOrionin ongelmana on monimutkaisuus. Siinä missä alkuperäinen Master of Orion loisti helposti ymmärrettävillä pelimekaniikoillaan ja yksinkertaisella käyttöliittymällä, FreeOrion tuntuu heti suorastaan päälleikäyvä vaikealta.

Käyttöliittymä on ulkoasultaan hieno ja sisältää monelle open source -pelille vierasta visuaalista karkkia. Pelattavuus kuitenkin jää kehnoksi käyttöliittymän bugien ja huonojen ratkaisujen takia. Erityisen piikikkäät risut ansaitsee monesta paikasta löytyvä opaslaatikko, jossa luetellaan kymmenittäin pelin kannalta tärkeitä asioita yhteen pötköön huonosti jaoteltuna.

FreeOrionin tarjoama pelikokemus jää muutenkin vajaaksi esikuvastaan. Alkuperäisen Master of Orionin taktinen avaruustais-telujärjestelmä loistaa poissaolollaan, mikä tekee myös laivaston suunnittelusta vähemmän mielenkiintoista. FreeOrion on tosin vielä alpha-vaiheessa – avointa Master of Orion -henkistä peliä metsästävän kannattaa pitää silmällä pelin kehitystä näiden puuttuvien ominaisuuksien varalta. Työn alla olevissa suunnitteludokumenteissa väläytellään mm. kolmiulotteista taktista taistelujärjestelmää.

Transport Tycoon

Julkaisija: MicroProse
Julkaisuvuosi: 1994

Simutrans

Projekti aloitettu: 1997
Ohjelmointikieli: C++
Lisenssi: Artistic License

OpenTTD

Projekti aloitettu: 2003
Ohjelmointikieli: C
Lisenssi: GNU GPL v2

Sid Meier's Colonization

Julkaisija: MicroProse
Julkaisuvuosi: 1994

FreeCol

Projekti aloitettu: 2002
Ohjelmointikieli: Java
Lisenssi: GNU GPL v2

Master of Orion

Julkaisija: MicroProse
Julkaisuvuosi: 1993

FreeOrion

Projekti aloitettu: 2003
Ohjelmointikieli: C++ ja Python
Lisenssi: GNU GPL v2 (moottori) ja CC-BY-SA 3 (pelidata)

Sid Meier's Civilization

Julkaisija: MicroProse
Julkaisuvuosi: 1991

FreeCiv

Projekti aloitettu: 1995
Ohjelmointikieli: C
Lisenssi: GNU GPL v2

Taistelu internetistä: vapauttajat vs rajoittajat

Internetin piti olla 2000-luvun Agora — demokraattinen kansalaisfoorumi, kulttuurien kohtaamiskeskus, joka tekisi maailmasta paremman paikan. Tuliko siitä isoveljen raippaa kaipaava anarkia, jossa vaaralliset ääriryhmät löytävät äänensä? Onko yksilön oikeus viestintäsalaisuuteen suurempi kuin sisällöntuottajan oikeus säädellä teostensa jakelua? Viranomaiset, edunvalvontajärjestöt, aktivistit ja poliittiset tahot pyrkivät vaikuttamaan historian monimutkaisimman teknisen järjestelmän toimintaan ja tulevaisuuteen.

Teksti: Eetu Korhonen

Kuvat: www.digital-delight.ch, Risto Mäki-Petäys, Eetu Korhonen, Mitol Berschewsky

Vapaus valvoa, vapaus valvonnasta?

Internetin tämän hetken keskeisin eettis-poliittinen kysymys koskee käyttäjän vapautta. Mitä maailmanverkon kansalainen saa ja ei saa tehdä? Kuka siitä päättää? Miten sääntöjä luodaan ja valvotaan ympäristössä, joka on hallitsemattoman laaja ja jatkuvasti muuttuva? Yritykset säännellä Internetin käyttöä ovat osoittautuneet helposti kierrettäviksi ja erittäin vaikeiksi valvoa. Torrent-trackerien sulkemiset, DNS-suodatusten kehittäminen ja kaistarajoitukset vaativat vuosien poliittista valmistelua, miljoonien eurojen laitehankintoja ja koulutusta uuteen toimintaympäristöön. Väärinkäyttäjälle – tai esimerkiksi poliittista sensuuria pakenevalle käyttäjälle – kaiken tämän ohittaminen tarkoittaa ainoastaan käyttötapojen hienoista muutosta esimerkiksi selainlaajennuksilla. Jos tämä ei riitä, motivoitunut käyttäjä voi vaihtaa palvelua heikommin valvottuun.

Nettiaktivistien palopuheissa korostuvalla tiedon valtatievapaudella on kääntöpuolensa, johon kontrollista innostuneet tahot tarttuvat kerta toisensa jälkeen. Vapaus toteuttaa itseään

tarjoaa yleensä mahdollisuuksia myös väärinkäyttöön. Netissä liikkuu valtava määrä kyseenalaista ja laitonta materiaalia aina kiusallisista haittaohjelmista järkyttävään lapsipornoon. Ei-toivotun materiaalin rajoittamista on yritetty niin teknisin, poliittisin kuin valistuksellisinkin keinoin. Tällaiset keinot nostavat kynnystä lähteä harhailemaan kaidalta polulta, mutta todella tehokkaita kieltoja tai edes jotakuinkin varmoja estoja Internetin käytölle ei voida saada aikaan puuttumatta ihmisten oikeuteen ylipäänsä käyttää tietotekniikkaa omaehtoisesti.

Kiltit lapset eivät lue kiellettyjä kirjoja

Ajankohtaisin aihe netin käyttöä koskevissa keskusteluissa on valtioiden, viranomaisten ja tiedustelupalvelujen rooli. Maailmanverkko ei ole kaikille sama, ja verkkosensuuria käytetään paikoitellen hyvinkin tehokkaasti poliittisten agendojen ajamiseen. Reporters Without Borders -järjestö listasi vuonna 2012 peräti 12 Internetin vihollisvaltiota, joissa verkkoliikennettä valvotaan ja suodatetaan poliittisen sensuurin nimissä. Estoista

tunnetuin on Kiinan suuri palomuuuri, joka sulkee maailman väkirikkaimman valtion verkkokansalaiset ulos hallituksen haitallisiksi katsomilta sivustoilta. Samankaltaisia maita ovat esimerkiksi Iran, Saudi-Arabia ja Valko-Venäjä, jotka kaikki suodattavat ja tarkkailevat verkkoliikennettään ja järjestelmällisesti häiritsevät poliittisten verkkoaktivistien toimintaa.

Suoraa valtiojohtoista sensuuria yleisempää on vaatia viranomaisia ja tietoliikennepalvelujen tarjoajia puuttumaan verkoissaan liikkuvaan laittomaan sisältöön. Verkkorikollisuuden vastaisessa työssä käytetään samanlaisia työkaluja kuin autoritääriset hallinnot käyttävät sensuuripuhissaan. Siksi lukuisat tahot kaupallisista suuryrityksistä pieniin aktivistijoukkoihin ovat huolestuneet käytetyistä keinoista ja niiden mahdollisista väärinkäytöksistä. Lähiaikojen menestyksekkäin hanke verkkokontrollin kasvun hillitsemiseksi lienee Yhdysvalloissa piratismiin kitkemiseen tarkoitettun SOPA/PIPA-lakialoitteen kaatamiseksi tarkoitettu tempaus. Sen myötä Googlen maineikas etusivudoodle muuttui sensuurivastaiseksi mustaksi laatikoksi, ja Wikipedia esti pääsyn englanninkielisiin sisältöihinsä. Lakialoite kumottiin, ja satojen miljoonien netinkäyttäjien arkeen vaikuttanut tempaus keräsi huomiota myös muun maailman verkkopaushankkeille. Huomattavin seuraus oli kansainvälisen ACTA-sopimuksen (Anti-Counterfeiting Trade Agreement) kaatuminen Euroopan Unionin käsittelyssä.

Jokapaikan tietotekniikan mahdollisuuksista huolestuneet verkkoaktivistit seuraavat suurennuslasilla armeijoiden ja tiedustelupalveluiden toimintaa netissä. Halpeneva tallennustila ja kasvava suorituskky tekevät yli kahden miljardin nettikäyttäjän yksilöllisen profiloinnin ja seurannan paitsi mahdolliseksi, myös todennäköiseksi turvallisuuspoliittisesti pehmeäksi tiedustelukeinoksi. Jo nyt Googlen ja Facebookin kaltaiset yritykset tekevät miljardibisnestä käyttäjiensä kulutustottumuksia ennakoimalla. Demokraattisten teollisuusvaltioiden tiedetään tarkkailevan ja tallentavan maansa sisällä liikkuvaa verkkoliikennettä, mikä kriitikoiden mielestä antaa liikaa valtaa ja mahdollisuuksia sitä valvoville viranomaisille. Mitä kotimaansa poliitikkojen vakoilusta kiinni jäänyt J. Edgar Hoover olisikaan nykyteknologialla saanut aikaan?

Viranomaisten toimivallan laajenemisen vastustajina toimivat vapaata Internetiä ja sen lieveilmiöitä puolustavat poliittiset, kansalais- ja aktivistijärjestöt. Näistä tällä hetkellä tunnetuimpia ovat Electronic Frontier Foundation EFF, kansainvälinen piraattiliike sekä häktivismiä poliittisena työkaluna käyttävät ryhmittymät kuten Anonymous. Internetin sääntelyn vastustamisen ohella toistuvia teemoja ovat esimerkiksi tekijänoikeus- ja patenttijärjestelmien uudistaminen sekä tietosuojan ja yksityisyyden rinnastaminen ihmisoikeuksiin.

Poltettiinhan Alexandriakin...

Internetin kehityksen suuntaan vaikuttavat myös kaupalliset toimijat. Tiettyjen olemassa olevien sääntöjen mukaan toimiva verkko on edellytys Googlen, Facebookin, Microsoftin ja Applen kaltaisten teknologiayritysten toiminnalle. Toisten liiketoimintamalleille kasvava Internet puolestaan muodostaa uhan. Kaksi tunnetuinta netissä etujaan valvovaa intressiryhmää ovat mediatalot ja tietoliikenneoperaattorit. Mediatalot ovat huolissaan aineettoman omaisuutensa

luvattomasta verkkojakelusta ja operaattorit maksullisten viestintäpalveluiden siirtymisestä Internetiin.

Marraskuussa 2012 Suomessa niin sanottu Chisugate nousi julkisuuteen, kun poliisi takavarikoi 9-vuotiaan tytön tietokoneen Tekijänoikeuden tiedotus- ja valvontakeskus TTVK ry:n tehtyä tutkintapyyntöön musiikin laittomasta verkkojakelusta. Aiemmassa Finreactorin tapauksessa puolestaan palvelun 21 ylläpitäjää tuomittiin noin 500 000 euron korvauksiin piraattilinkkien tarjoamisesta ja teosten laittomasta jakamisesta. Vertaisverkkopalvelut ovatkin luoneet uudenlaisia haasteita piratismiin vastaiselle työlle, sillä Napsterin jälkeisessä maailmassa käytännössä jokainen laittomien tiedostojen lataaja on samalla myös palvelun toiminnan mahdollistava tiedostojen jakaja.

Toimet Piratebayta ja muita julkisia latauslinkkejä jakavia piraattitrackereitä vastaan ovat siis osoittautuneet ongelmallisiksi. Mediatyhtiöiden ja artistien etuja ajavat toimijat ovat siirtyneet uhkaamaan myös aiemmin koskemattomuudesta nauttineita tiedostojen lataajia, tavallisia käyttäjiä. Laiton lataaminen on tällä hetkellä jotakuinkin yhtä yleinen rike kuin punaisia päin kävely, eikä piraattipalvelimien ylläpitäjiä ja massarikollisia vastaan suunnattu rangaistumennettely näin suuressa mittakaavassa toimi. Oikeuspoliittisen tutkimuslaitoksen mukaan noin 70% 15-16 vuotiaista nuorista on käyttänyt laittomia latauspalveluita. Heistä lähes jokainen syyllistyy myös materiaalin jakamiseen. Tekijänoikeusjärjestöt pohtivatkin tällä hetkellä useita ”pehmeitä” lakeja yksityislataajien valistamiseksi ja rankaisemiseksi.

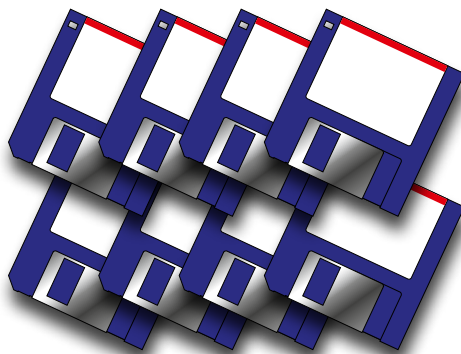
Tekijänoikeuskysymyksissä nousee esiin ristiriita perinteisten mediatyhtiöiden ja uusien, tiedon jakamista ja saatavuutta edistävien teknologiayhtiöiden välillä. Mediaimperiiumi News Corp.:in pääomistaja Rupert Murdoch nimesi Googlen ”piratismijohtajaksi” ja varkaaksi Googlen liittyttyä SOPA- ja PIPA-lakihankkeiden vastustajiin. Googlen ja Youtuben kaltaiset palvelut indeksoivat ja tallentavat tekijänoikeudella suojattua aineistoa ilman lupaa, eivätkä yleensä pysty estämään käyttäjiensä pääsyä materiaaliin. Esimerkiksi Youtube on myöhemmin ryhtynyt tilittämään tunnistamistaan videoista rojalteja niiden oikeudenhaltijoille. Monet lailliset mediapalvelut ovat joutuneet hintakilpailuun, mikä ei miellytä kaikkia sidosryhmiä. Tunnetuin tapaus lienee Spotifyn mahdollinen vaikutus levymyyntiin ja artisteille tilittettävät rojalit, mitä monet tekijänoikeustuloista elävät muusikot ovat kritisoineet.

Taistelussa materiaalin vapaata leviämistä vastaan on esitetty myös käyttörajoituksia. Useimpien suomalaisen muistissa lienee vielä vuoden 2005 Lex Karpela, joka pyrki kieltämään tehokkaiden kopiosuojausten kiertämiseen tarkoitettujen ohjelmistojen ja ohjeiden jakelun. Saksassa puolestaan kiellettiin vuonna 2007 tietoturvan kiertämiseen tarkoitettujen työkalujen hallussapito, kehittäminen ja jakelu. Toistaiseksi tällaiset lait ovat osoittautuneet valvomiskelvottomiksi kuolleiksi pykäliksi, joiden turhan tiukka tulkinta on aiheuttanut huolta muun muassa kaupallisten tietoturvapalveluiden tarjoajien parissa.

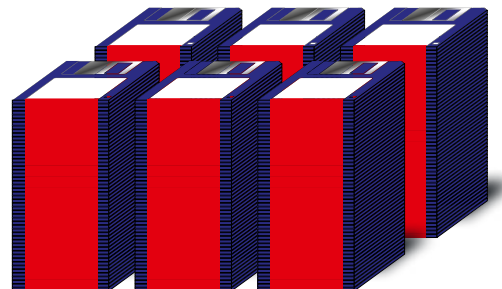
Eurolla voit siirtää...



Puoli sivua tekstiviestejä, 0,08 €/kpl



12 minuuttia puhetta, 0,08 €/min



1,5 tuntia videopuhelua dataliittymällä, 12 €/3Gt

Paljonko maksaa?

Tietoliikenneoperaattorien tarjoamien palveluiden hinnoittelu ei perinteisesti ole riippunut palvelun tarjoamisen kustannuksista. Esimerkiksi SMS-viestien lähettäminen on käytännössä merkityksetöntä matkapuhelinverkon kuormitukselle: minuutti puhetta vaatii tyypillisellä pakkausmenetelmällä noin puolen megatavun siirtoa, kun tekstiviestin 160 merkkiä siirtyy alle tuhannesosassa tästä. Internetissä eri palvelutyyppä ei voida luontevasti tunnistaa ja erottaa toisistaan. Tekstiä, videokuvaa ja puhetta voidaan siirtää niin selaimen TCP- ja HTTP-protokollien päällä, kuin myös erillisillä pääteohjelmistoilla, yhteydettömän tiedonsiirron avulla tai vertaisverkkoja pitkin. Operaattori voi hallita tästä kaikesta lähinnä tietokoneita ohjaavaa reititystä ja verkon fyysistä ylläpitoa.

Koska puheliikenne, pikaviestit ja kaapelitelevisiön jakelu ovat kasvavassa määrin siirtymässä julkiseen Internetiin muiden toimijoiden haltuun, ovat operaattorit huolissaan oman liiketoimintansa kaventumisesta. Tätä rajoittaakseen monet operaattorit ovat päätyneet kannattamaan kaiken verkkoliikenteen samanarvoisena näkevän network neutrality -periaatteen poliittista ja teknistä rajoittamista.

Tämä mahdollistaisi tietoliikennepalvelujen tarjoajalle tilaisuuden käyttää teknisiä keinoja sisällön priorisoimiseksi ja suodattamiseksi. Tänä päivänä tyypillinen rajoite voi esimerkiksi puuttua verkossa tapahtuvan BitTorrent-tiedonsiirron kaistanleveyteen tai varmistaa VoIP- ja IPTV-palvelujen tarvitseman kaistanleveyden saatavuuden riippumatta yhteyden rinnakkaisesta käytöstä. Jyrkän network neutrality -tulokannan mukaan näin ei tulisi saada tehdä, vaan jokaista yhteyttä tulisi kohdella samanarvoisesti sen sisällöstä, toimintaperiaatteesta ja tarjoajasta riippumatta.

Teleoperaattoreiden toimintaan vaikuttavat monet risteävät viranomais-, kansalais- ja kaupalliset intressit. Monet palveluntarjoajat ovat päätyneet puoltamaan kansallisvaltioiden pyrkimyksiä loppukäyttäjien tiedonsiirron suodattamiseksi. Esimerkiksi piratismien kitkeminen voi hyvin olla myös videovuokrausta ja netti-tv:tä tarjoavien verkkotoimijoiden kannalta edullista. Toisaalta se tuottaa uudenlaisia kustannuksia, vastuuta ja rajoituksia liiketoiminnalle, mistä tuoreimpina esimerkkeinä Elisan oikeudenkäynnit Pirate Bayn DNS-suodattamisen estämiseksi ja IPTV:n 36 tunnin katselupuskurin poistosta.

Tee itte parempi!

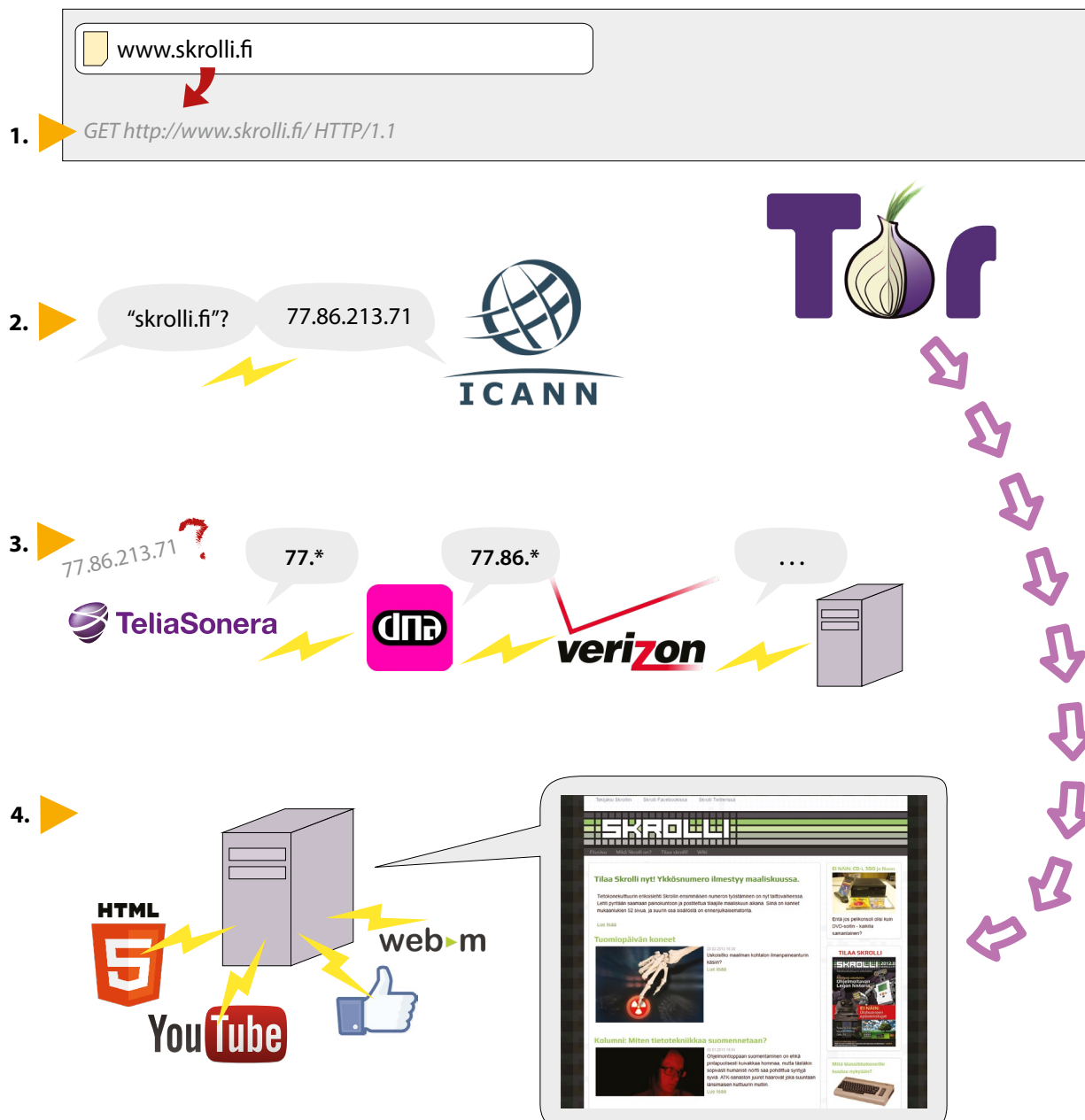
Niin suuryritysten, poliittisten toimijoiden kuin tiedustelupalveluidenkin toimintaa tarkastellessa on syytä muistaa, että Skype, Facebook, Google, Amazon, Linux, GNU ja lukemat-

tomat muut verkon peruspalvelut ovat syntyneet yksittäisten käyttäjien toimesta. Dystooppisimmatkin valvontakoneistot ovat tähän mennessä olleet lähinnä reaktioita odottamattomiin käyttötarkoituksiin, joita pioneerihenkiset hakerit ovat kehittäneet. Ei ole mitään syytä olettaa, etteikö visionäärien mielikuvitus olisi jatkossakin tärkein Internetin kehitystä ajava voima.

Vaikka Internet ei tarvitse keskushallintoa toimiakseen, kaksi voittoa tavoittelematonta ja monin osin vapaaehtoisuuteen perustuvaa järjestöä ovat nousseet merkittävimiksi verkon teknistä toteutusta ja yhteensopivuutta kehittäviksi tahoiksi. Yhdysvaltalaisen ICANNin (Internet Corporation for Assigned Names and Numbers) vastuulla on pitää kirjaa käytössä olevista verkko-osoitteista ja hallinnoida näihin ohjaavan nimipalvelinverkon (DNS) juuripalvelinten toimintaa. ICANNin monopolia on vastustanut esimerkiksi YK:n Kansainvälinen televiestintäliitto, samoin kuin monet kilpailijat kaupallisista osoitepalveluista anonyymiin TOR-darknetiin.

Internet Society puolestaan toimii kattojärjestönä useille pitkäikäisille asiantuntijaelimille ja think tankeille, joiden tarkoituksena on edistää tietoverkon teknistä kehitystä ja asemaa yhteiskunnassa. Internet Societyn elimistä mainittakoon Internet Engineering Task Force (IETF), jonka julkisista RFC-pohdinnoista (Request For Comments) muodostuvat Internetin teknistä toteutusta kuvaavat standardit, sekä Internet Research Task Force (IRTF), joka pyrkii kartoittamaan Internetin evoluution ja muuttuvien tarpeiden tulevia polkuja.

Internet-aktivistien puheissa nauretaan usein yrityksille rajoittaa ja valvoa verkon käyttöä. Maailma on pullollaan syntyessään kuolleiksi haukuttuja IP-suodattimia, maksumuureja ja viranomaisia sitkeästi väisteleviä verkkopalveluja. Aukottomia tai edes kovin hyviä rajoittimia lienee mahdotonta tehdä niin kauan kuin Internetin perusinfra pysyy pääosin avoimena. On kuitenkin syytä muistaa viisaus "valmis on parempi kuin täydellinen." Poliitikko on kompromissien etsimistä, ja huonokin toteutus on usein arvokkaampaa kuin täysi toimetttömyys. Uraan laajuus ei yrittäjiä lannista, eikä mitättömän oloisia saavutuksia kannata vähätellä. Toimivan hallinnon vaatima tekninen kynnys on myös käyttäjää vastassa. Väinö Vainoharha on toki vapaa siirtämään arkiselailunsa darknetiin, maksuliikenteensä Bitcoinin ja viestintänsä vahvan salauksen piiriin, mutta käytännössä erittäin harva päätyy tekemään näin. Yksinkertaiset ja helppokäyttöiset palvelut ovat yleensä suosituimpia. Eikä WWW:tä, Facebookia ja sähköpostia kukaan haluaisi hallita, elleivät käyttäjät muodostaisi niistä helppoja ja houkuttelevia maaleja..



Erään verkkopyynnön elinkaari

1. Selain muuntaa käyttäjän sivupyynnön IETF:n ja W3C:n standardoimaksi HTTP-kutsuksi. Uusi HTTP 2.0 on työn alla, jonka mahdollisia prototyyppejä ovat tarjonneet mm. Google ja Microsoft.

2. Selain utelii DNS-verkolta skrolli.fi-domainin takana olevan verkkopalvelimen IP-osoitetta. Tavallisimmat DNS-palvelimet ovat yksittäisten verkko-operaattorien ylläpitämiä, mutta niiden yhteistoimintaa koordinoivat juuripalvelimet ovat yhdysvaltalaisen ICANNin hallinnassa.

Toisinaan oikeuslaitokset määräävät DNS-palvelimia estämään osoitetietojen antaminen laittomiksi katsottuihin palveluihin. Teknisesti tämä on hyvin tehoton estokeino, koska käyttäjä voi halutessaan vaihtaa käyttämänsä DNS-palvelimen ulkomaiseen.

3. Pyyntö lähetetään reititysverkolle, joka ohjaa pyynnön ja sille saatavat vastaukset lähemmäs kohdetietokoneita. Fyysistä verkkoa ja reititystä ylläpitävät tavallisesti teleoperaattorit.

IP-reititystä on käytännössä mahdotonta täysin sivuuttaa nykyisessä Internetissä, mutta TORin kaltaisten anonymisointipalveluiden avulla tosiasiallinen polku koneiden välillä voidaan sekoittaa tunnistamattomaksi.

4. Kutsu ohjautuu kohdepalvelimelle, joka vastaa pyyntöön pyydetyllä sisällöllä. Yleensä kohdepalvelimet hyödyntävät liitua valmiita verkkosovelluksia, joihin kohdistuvat muutokset vaikuttavat täten suoraan myös jalostettuun palveluun.

Tuomiopäivän koneet

Uskoisitko maailman kohtalon ilmanpaineanturin käsiin?



Vaikka se nostalgiahuuruissa välillä unohtuukin, 1980-luku ei ollut pelkkiä varsilenkkareita, Ihmemiestä ja New Orderia. Etenkin vuosikymmenen alkuvuosina ilmassa leijui erittäin todellinen uhka suurvaltojen välisestä konfliktista, joka päättyisi ydinaseiden käyttöön. Sen seuraukset taas olivat tiedossa: elämä loppuisi ainakin siinä muodossa, kuin sen länsimaissa käsittämme. Kasarilapselle tämä ajatus ei ollut iltaisin erityisen rauhoittava, ja vielä huonommin olisi nukkunut tietäessään, miten lähellä täysimitaista ydinsotaa oikeasti käytiin.

Teksti: Mikko Heinonen

Kuvat: Manu Pärssinen,

Wikimedia Commons



Sekä Yhdysvallat että Neuvostoliitto samanaikaisesti pelkäsivät toisiaan ja olettivat, ettei vastapuoli mitenkään voi kuvitella heidän olevan se taho, joka "painaisi nappia" ensimmäisenä. Tämä absurdus saavutti yhden huippupisteensä NATOn Able Archer 83 -sotaharjoituksen aikana. Länsimaat harjoittelivat tilannetta, jossa ydinsota on alkanut, jolloin NL:n vakoilu luuli, että ne ovat aloittamassa ydinsotaa harjoituksen varjolla. Agentteja pyydettiin raportoimaan vain havaintonsa, ei päätelmiään, ja kun jokainen näki kokonaisuudesta vain pienen osan, soppa oli valmis. Tilanne laukesi käytännössä vasta, kun sotaharjoitus päättyi.

Sotaa valkokankaalla

Suurvaltojen kireät välit vaikuttivat luonnollisesti myös populaarikulttuuriin, ja varsinaisten synkkien tuomiopäiväkuvausten (Testament, Threads, The Day After) lisäksi ydinsota vilahteli aiheena muunkinlaisissa elokuvissa. Kuuluisimpia näistä on vuoden 1983 Wargames, jossa nuori hakkeri David (Matthew Broderick) tulee pelejä etsiessään vahingossa soittaneeksi ydinaseita hallinnoivalle tietokoneelle. Vaikka Hollywood-tyyliin mutkia onkin oiottu, elokuvassa esitetään oikeasti sovellettavissa olevia keinoja salasanojen selvittämiseen. Se

on toiminut monelle innoittajana tietokoneharrastukseen.

Eräs päähenkilöistä on professori Falckenin (John Wood) luoma WOPR-keinoäly (War Operation Plan Response), joka pyrkii simuloimaan erilaisia konfliktitilanteita ristinnollasta maailmansotaan asti. Se otetaan käyttöön, koska ohjusiloissa istuvat ihmiset ovat haluttomia laukaisemaan ydinaseensa selvästä käskystä huolimatta. David tulee vahingossa käynnistäneeksi WOPRin "pelaamaan" ydinsotaa, jota sotilasjohto luulee oikeaksi. WOPR yrittää myös itse laukaista ohjukset, ennen kuin kauniin optimistisesti päättyy toteamaan ydinsodan olevan peli, jonka voi voittaa vain olemalla pelaamatta sitä. Ilmeiseksi käy myös, että koneen harkinta pettää ainakin yhtä herkästi kuin ihmisen.

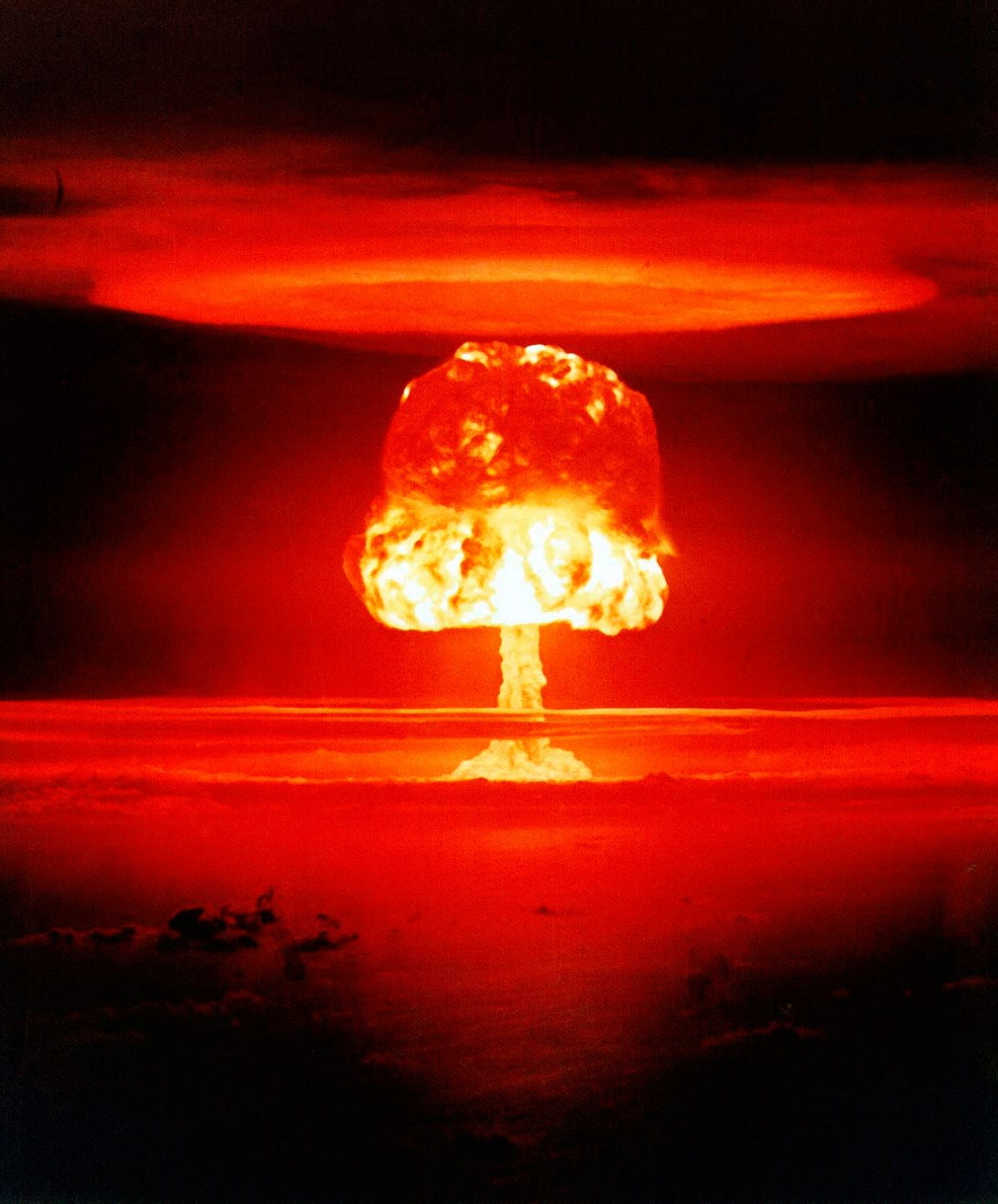
Elämä matkii taidetta

Näin tulevaisuudesta katsottuna Wargamesin yhtäläisyydet tosielämään ovat runsaampia kuin käsikirjoittajat ehkä tulivat ajatelleeksikaan. Elokuva perustui löyhästi 80-luvun alussa sattuneeseen tapaukseen, jossa Yhdysvaltain armeijan ydinpommittajat olivat jo ilmassa komentokeskus NORADista saatujen virheellisten tietojen vuoksi. Syynä ei ollut pelejä etsinyt hakkeri vaan "sotasimulaatio-tilaan" jäänyt tietokone, jonka viestejä

luultiin oikeiksi. Tapauksia oli ollut ainakin kaksi muutakin, joista toisessa kyse oli inhimillisestä virheestä, toisessa jälleen kerran tietokoneen toimintahäiriöstä. Tällöinkin osa aselajeista jätti toimimatta ohjeiden mukaisesti, koska katsoi hälytyksen olevan varmasti väärä. Sama tieto oli tuossa vaiheessa saavuttanut jo muutkin, joten todellinen maailmanpalon uhka oli jälkikäteen arvioituna pieni.

Vuoden 1983 syyskuussa oltiin kuitenkin Atlantin toisella puolen tilanteessa, jossa inhimillinen harkinta esti mahdollisesti vakavatkin seuraukset. Upseeri Stanislav Petrov istui valvomassa Neuvostoliiton valvontajärjestelmän signaaleita, kun hän sai ilmoituksen NL:ään suunnatun Minuteman-ohjuksen laukaisusta. Harkittuaan asiaa hetken Petrov kuittasi viestin vääräksi hälytykseksi.

Ei ehtinyt kulua kauan ennen kuin järjestelmä piippasi uudelleen, tällä kertaa neljästä ohjuksesta. Petrov arveli, että sodan oikeasti alkaessa Yhdysvalloista ei tulisi muutamaa, vaan satoja ohjuksia kerralla, ja hylkäsi tämänkin ilmoituksen. Päätelmä oli rohkea mutta oikea, ja laitteen toimintahäiriö todettiinkin jälkikäteen. Jättämällä järjestelmän signaalin huomiotta Petrov ei ehkä suorastaan ehkäissyt maailmanloppua, mutta ainakin teki osansa konfliktin välttämässä. Paljolti internetissä levinneen



tiedon ansiosta hän onkin noussut jonkinlaisen legendan asemaan.

Tämän tiedon pohjalta kaikkein mielenkiintoisin – tai sitten kauhein – yksityiskohta onkin se, että eräänlaisen WOPRin huhutaan sitkeästi olevan olemassa. Ei kuitenkaan Yhdysvaltain NORADissa, vaan Kremlissä.

Kuollut käsi kosta

Sistema Perimetr, jota länsimaissa on kutsuttu myös nimellä Dead Hand, oli osa Neuvostoliiton ydinpuolustusta. Tiedot järjestelmän todellisista ominaisuuksista, olemassaolosta, toimintavalmiudesta ja etenkin automaation määrästä vaihtelevat lähteestä toiseen, mutta perusajatus on kaikkien mukaan samansuuntainen. Saataessa kohtalaisen varmoja tietoja siitä, että maahan on kohdistumassa ydinasehyökkäys, voidaan päätäntävalta siirtää Perimetrille. Se tarkkailee ympäristön valon määrän, säteilytason ja ilmanpaineen muutoksia sekä mahdollisesti myös radiolähettämiä. Niiden perusteella järjestelmä tekee pää-

töksen siitä, onko sitä vastaan hyökätty. Jos näin on käynyt, Perimetr laukaisee jäljellä olevan ydinasearsenaalin.

Perimetr eroaa elokuvan WOPRista siinä, että se on selkeästi kostoase. Samalla se on itse asiassa suunniteltu ylläpitämään rauhaa, ei suinkaan tuhoamaan ihmiskuntaa. Kun on olemassa teoriassa erehtymätön taho, jolle voidaan siirtää lopullinen päätös ydinasehyökkäyksen tekemisestä, kenenkään ihmisen ei tarvitse tuota päätöstä tehdä.

Näin vältetään se mahdollisuus, että maailma palaisi jonkin yhden tai useamman henkilön virhearvion vuoksi. Jos Neuvostoliitto olisi yllätetty ydinaseilla, kosto olisi ollut varma riippumatta siitä, mitä politbyroo päättää. Osaltaan tämä vähentäisi myös Able Archer -tyylisiä väärinkäsityksiä.

On vaikeaa löytää luotettavaa tietoa siitä, millainen Perimetr oikeasti on, missä määrin se on toiminnassa ja onko se koskaan ollut automaattinen. Wired-lehden vuonna 2009 julkaiseman artikkelin mukaan järjestelmä on toimin-

tavalmiudessa ja sitä päivitetään edelleen, mutta entiset neuvostokenraalit taas sanovat ettei järjestelmää koskaan käynnistetty. Ristiriitaisen tiedon syykin on ilmeinen: Venäjälle on hyödyllistä ylläpitää mielikuvaa siitä, että sillä on käytössään ainutlaatuinen kostojärjestelmä mahdollisten hyökkääjien varalta. Muille maille on hyödyllistä pitää Venäjää hiukan pelottavana, vaikei siitä samantyyppistä pahan valtakuntaa saisikaan kuin edeltäjästään. Totuus asiasta selvinnee vasta aikojen kuluttua, jos silloinkaan.

Palataan kuitenkin hetkeksi Stanislav Petroviin. Hän oli rajapinta koneen ja ympäristön välissä ja täytti tehtävänsä torjuessaan virheellisen ilmoituksen. Mutta jos Petrovin sijasta penkillä olisi istunut Perimetr, kuka tietää mitä olisi tapahtunut. Tämä yhtälö oli epäilemättä tiedossa Perimetrin suunnittelijoilakin, joten on vaikea uskoa, että kukaan olisi jättänyt maailman tuhoamiseen tarvittavan asemäärän tunnetusti epäluotettavan tekniikan hallintaan. Todennäköisempää on, että tuotantokäyttöön tehdyssä järjestelmässä olisi ollut omat Petrovinsa istumassa maanalaisissa siiloissaan.

Isännästä rengiksi

Kylmä sota on onneksemme päättynyt, eikä maailmanlaajuinen ydinsota enää vaikuta kovinkaan todennäköiseltä skenaarilta. Kukaan tuskin enää tosissaan ehdottaa, että ohjukset pitäisi alistaa ”erehtymättömän” tietotekniikan valvotavaksi.

Ydinaseita on kuitenkin maailmassa edelleen enemmän kuin tarpeeksi, ja tietotekniikka on niihin naimisissa entistäkin vahvemmin. Koska ydinkokeita ei enää tehdä, tarvitaan aseiden kehitykseen mallintamista ja laskentaa yhä enemmän. Samoin ohjausjärjestelmät ovat kehittyneet jatkuvasti.

Kun DEC esitteli 64-bittisen Alpha-prosessorinsa 90-luvun alussa, sen ohjelmistotuki oli aluksi vaatimatonta. Pahat kielet ehtivät huhuilemaan, että se ei Digitalia huolestuta, koska risteilyohjuksessa ei tarvitse ajaa kummoista käyttöjärjestelmääkään. Tämän perustaksi ei löydy varmaa tietoa, mutta sen sijaan faktaa on se, että eräs suurimpia Alpha-pohjaisia supertietokoneita asennettiin vuosituhaten vaihteessa Los Alamosin laboratorioon – juuri siihen, jossa maailman ensimmäiset ydinaseet kehitettiin.

EI NÄIN!

Atari Jaguar, kissapedon kuolonkorinat

Vuoden 1990 tienoilla Atarilla ei taas kerran mennyt kovin hyvin. Firma, joka oli kerran ollut pelaamisen synonyymi, oli kaatanut maahan jo monta mahdollisuutta menestykseen.

Mikko Heinonen

Kuva: Manu Pärssinen

Amerikan pelaamaan opettanut Atari 2600 -pelikonsoli oli käytännössä romahduttanut koko toimialan 10 vuotta aiemmin, kun kuluttajat kyllästyivät sen laaduttomiin peleihin. Samassa yhteydessä oli mennyt nurin sopimus Nintendon konsolien edustuksesta USA:ssa.

Romahduksen jälkeen Atari oli jaettu kahtia, Atari Gamesiin ja Atari Corporationiin. Atari Games oli tehnyt eräitä 1980-luvun parhaista kolikkopeleistä, kun taas Atari Corporation oli keskittynyt julkaisemaan vanhoja laitteita (2600, 7800, XL, XE) uudelleen. Hävittyään taistelun Amigasta se oli kehittänyt nopeasti Atari ST:n, joka oli kohtalainen menestys ja sai jopa myöhemmin seuraajikseen TT:n ja Falcon030:n.

Epyx-ohjelmistotalon kassakriisistä pelastettu käsikonsoli Atari Lynx puolestaan tuli markkinoille 1989. Vaikka sen pelit olivat keskimäärin hyviä, laite oli kallis ja söi järjettömästi paristoja. Se ei ehtinyt saavuttaa kriittistä massaa, ennen kuin Nintendo Game Boy tuli ja tyhjensi pajatson.

Varaa epäonnistumisiin ei enää ollut. Uuden kotikonsolin piti olla menestys, joten sitä tekemään valjastettiin Flare Technology. Yritys oli tätä ennen tehnyt mullistavan Multisystem-pelikonsolin. Tai siis konsoli olisi ollut mullistava, jos sen kehityskustannukset eivät olisi kaataneet valmistaja Konixia ennen kuin yhtään laitetta ehdittiin myydä.

Ataria Flare Technologyn aiemmat

ongelmat eivät pelottaneet, vaan työt aloitettiin kahden uuden kissaeläimen parissa. Suunnittelupöydällä olivat 32-bittinen Panther ja 64-bittinen Jaguar. Pian Panther sai kirveestä, sillä sen Motorola MC68000 -pohjainen arkkitehtuuri ei enää vuonna 1991 ollut riittävän vakuuttava.

Jaguarin sen sijaan piti lyödä kilpailijansa Sega MegaDrive ja Super Nintendo kaikilla rintamilla. Luvassa oli 64-bittinen konsoli, jonka rauta tuki suoraan esimerkiksi Gouraud-varjostusta. Laitteessa oli muitakin teknisesti edistyneitä jippoja, joista 16-bittisillä alustoilla saattoi vain haaveilla.

Joulukuksi 1993 esitelty Jaguar ei kuitenkaan todellisuudessa ollut aivan sitä, miltä se näytti. Koneen 64-bittisyys koski vain grafiikkapiiriä. Suorittimena toimivat kaksi 32-bittistä RISC-yksikköä. Monimutkaisessa paletissa oli mukana myös vanha MC68000, jonka piti toimia vain "väyläohjaimena". Totuus kuitenkin oli, että Atarin kehitystyökalujen puutteen sekä arkkitehtuurin yleisen sekavuuden vuoksi useat Jaguar-pelit itse asiassa käyttivät tätä väyläohjainta pelikoodin pyörittämiseen.

Tämän seurauksena Jaguarin karjailu muistutti pikemminkin naukumista. Sille ilmestyi muutama kohtalainen peli, mutta valtaosa noin 60 nimikkeestä oli hätäisesti kyhättyä sotkua. Tämän sain kokea myös itse erehdyttyäni Jaguarin ostamaan. Enemmän olisi harmittanut vain, jos olisin ehtinyt hankkia täyteen

hin-
taan myös
laitteen hyödyt-
töman CD-aseman.

Eräs kuvaava tapaus oli, kun Jaguar-konsolin julkaisun yhteydessä Jaguar-autojen valmistaja otti Atariin yhteyttä. Autotehdas toivoi hyvässä hengessä yhtä konsolia voidakseen tutkia, ettei laite riko heidän tavaramerkkiään. Penneistään tarkka Atari-pomo Sam Tramiel otti PR-pisteet kotiin lähettämällä videokasetilla kuvaa Jaguarin peleistä. Ehkä juuri tämä asenne auttoi myymään peräti 125 000 Jaguaria, siinä missä MegaDrive myi 40 miljoonaa ja SNES vielä enemmän.

Jaguarin epäonnisessa tarinassa on kuitenkin yksi voittaja: homebrew-kehittäjät. Kun väistämättömän konkurssin jälkeen Atarin jäänteet päätyivät Hasbrolle, julisti se Jaguarin olevan heidän puolestaan vapaata riistaa. Sille ilmaantuikin peräti muutama kaupallinen julkaisija, joiden pelejä saa edelleen ostettua ja satunnaisia julkisohjelmiakin on tipahdellut tähän päivään asti. Kerrankin voi myös rehellisesti sanoa, että harrastelijoiden tuotokset ovat keskimäärin laadukkaampia kuin viralliset julkaisut.

Konkurssin yhteydessä Jaguarin kuorimuotit myytiin yritykselle, joka präsäsi niillä hammaslääkärin hoitolaitteita. Sellaisella suoritettu toimenpide olikin varmasti useimmille se miellyttävän ja antoisin kokemus Jaguarin parissa.



Vanhojen bittien pelastajat

Aki Sivula, Jarkko Lehti

Kuvat: Manu Pärssinen

80-luvun tietokonepelit, demot, introt ja ohjelmat ovat jättäneet lähtemättömän jälkensä tuolloin lapsuuttaan ja nuoruuttaan eläneiden mieleen. Aikakauden digitaalinen taide ja kulttuuri tukkifontteineen sekä palikkagrafiikkoinen tallennettiin tuolloin levykkeille ja kaseteille. Kyseisten medioiden lyhyen elinkaaren vuoksi niille tallennettu materiaali on kuitenkin katoamassa hitaasti mutta varmasti bittien taivaaseen. Kadotus ei kuitenkaan ole väistämätön kohdalo, vaan arkistoinnilla voidaan pelastaa tärkeä osa digitaalisen kulttuurin varhaisvaiheesta.

Digitaitteeksi voidaan lukea ainakin demot, introt, pelit, musiikki ja grafiikka. Myös lähdekoodit, tekstitiedostot ja hyötyohjelmat ovat arvokkaita historiallisia artefakteja ja aikansa kuvia. Tuolloin suosittujen alustojen arkistointi on kohdallaisen hyvällä mallilla, mutta harvinaisempien alustojen materiaalia ei vielä arkistoida kovin aktiivisesti, puhumattakaan hyötyohjelmista, joiden säilytysarvo nähdään usein olemattomana.

Aikanaan ei tullut mieleenkään, että Commodore 64:n klassikkopelit voisivat olla vaarassa kadota maan päältä, puhumattakaan Finnish Gold -skeneryhmän introista tai Tristar & Red Sector Inc:n Amiga-demoista.

Piraattiversioita peleistä löytyy netistä, mutta kuten tavallista, ne ovat usein puutteellisia, huonosti murrettuja tai ne on tehty esimerkiksi pelkästä kasettiversiosta. Lisäksi monista peleistä puuttuvat crack-versiot kokonaan, ja samannimisten pelien EU- ja USA-versiot eroavat toisistaan johtuen PAL- ja

NTSC-laiteversioiden sekä markkinoiden eroista.

Digitaitetta on arkistoitu koko sen olemassaolon ajan, ja useimmiten kyse on ollut kotikutoisesta puoliammattilaisten puuhastelusta. Esimerkiksi 90-luvulla BBS-skenessä törmäsi harrastajien tekemiin kokoelmiin, joille oli kerätty pelejä, demoja tai musiikkia. 2000-luvulla Internet mahdollisti ensimmäiset kunnolliset arkistointiprojektit. Näistä esimerkkinä pitää mainita lähes kaikkea mahdollista materiaalia kokoava TOSEC, Amiga-moduuleihin erikoistunut Amiga Music Preservation, alkuperäisten Commodore 64 -pelien arkistointiprojekti Commodore 64 Tape Preservation ja skenemateriaaliin erikoistunut CSDb. Näiden lisäksi löytyy lukuisia muita harrastelijoiden ylläpitämiä projekteja, joiden tavoitteet ja arkistointiaste vaihtelevat.

SPS – Software Preservation Society

Mikä sitten ei ole puuhastelua? SPS on Isosta-Britanniasta käsin toimiva yhdistys, joka ottaa arkistoinnin tosissaan. SPS arkistoi diskettimuodossa olevia pelejä tai peleihin verrattavia tuotteita, joita ei enää saa alkuperäisiltä toimittajilta. Yhdistys tunnettiin aikaisemmin nimellä CAPS (Classic Amiga Preservation Society), ja se keskittyikin nimensä mukaisesti Amiga-pelien arkistointiin. Toimintaa laajennettiin koskemaan myös muita alustoja, ja nykyisin SPS ottaa vastaan disketille tallennetut kaupalliset pelit joita ei ole aiemmin arkistoitu. Tällä hetkellä SPS:n tukema arkistointi kohdistuu mm. Amigan, Atarin, Spectrumin ja Commodore 64:n alkuperäisiin disketti-

peleihin.

Yhdistyksen päätavoite on taata pelien luotettava arkistointi. Tällä tarkoitetaan tuotteiden autenttista tallentamista sellaisina, kuin ne ovat tuotannosta tai kaupasta hankittaessa olleet. Arkistoitujen tuotteiden tulee olla virheettömiä, ja median sisällön on oltava siinä kunnossa, kuin se on tuotannosta tullessaan ollut. Sisällön muutokseksi lasketaan esimerkiksi pelien tallentamat high score -listat tai käyttäjän tekemät muutokset. Pitkän tutkimus- ja tuotekehitystyön tuloksena SPS:llä on käytössään tietoa ja teknologiaa, joka mahdollistaa tasokkaan ja luotettavan arkistoinnin. Nykyisin arkistointi käsittää myös peleihin liittyvää oheismateriaalia kuten alkuperäisiä lähdekoodeja tai kansitaitetta.

Kaupallisten pelien valtavasta määrästä johtuen SPS luottaa vapaaehtoisten työpanokseen. Yhdistyksen toimintaa tukevat muun muassa tietyt museot ja arkistointiprojektit. Lisäksi tähtäimessä on pelien laillinen uudelleenjulkaisu, kunhan tekijänoikeuksiin liittyvät kysymykset on ratkaistu.

Disketit ja kasetit talteen

Pelien ja demojen kopiointi oli alusta saakka yleistä huvia, niinpä kopiosuojauksen kehittäjien ja kräkkereiden kilpajuoksu alkoi jo varhain. Usein entistä parempia kopiosuojauksia saattoivat kehittää samat henkilöt jotka aiemmin kräkkäsivät pelejä. Commodore 64:llä Epyx käytti levyjen suojaukseen Vorpalia, Accolade Rapidlokiä, Mirrorsoft V-Max!:n variantteja ja niin edelleen. Kuriositeettina mainittakoon, että Vorpalin tarkka toiminta selvitettiin vasta 2011, jolloin



C64-peleistä saatiin ensimmäiset viralliset jäljennökset.

Tällä hetkellä ainoastaan SPS:n arkistointimenetelmä tallentaa pelit viimeistä bittiä myöten. Kaikki jäljennetty materiaali tarkastetaan, ja tarvittaessa otetaan useita jäljennöksiä, jotta voidaan varmistua, että levyke on 1:1 alkuperäisen kopio. Diskettien ammattimainen monistaminen oli jo 80-luvun alussa tarkkaan organisoitua bisnestä varsinkin USA:ssa. Levyjä kopioitiin duplikaattoreilla, jotka olivat levyasemista rakennettuja usean levyaseman monistuskoneita. Niillä pystyttiin tuottamaan kopiosuojauksilla varustettuja kaupallisia kopioita varsin tehokkaasti. Niiden avulla levyille kirjoitettiin lisäksi metadataa, jota ei pystytty esimerkiksi Commodoren 1541-asemalla lukemaan. Nykyään kyseinen metadata toimii yhtenä osana jäljennöksen autenttisuuden varmistuksessa. Pitää kuitenkin

muistaa, että muun materiaalin kuten Skenediskettien arkistoinnissa riittää varsinaisen datan arkistointi koska levyjä ei ole ammattimaisesti monistettu eikä metadataa näin ollen ole.

Kuinka diskettien digitaiteen tallennus sitten tapahtuu käytännössä? Eräs hyvä vaihtoehto on Kryoflux, joka tukee useimpia nykyisin käytössä olevia käyttöjärjestelmiä. Laite on USB-porttiin liitettävä ohjain, jolla voi ohjata lähes mitä tahansa levyasemaa. Käytännössä useimmat 3,5" korppuasemat ja 5,25" lerppuasemat sopivat vanhan median lukemiseen, ja tuki löytyy myös harvinaisemmalle 8" levyasematyypille. Ohjaimen omistajaksi pääsee vajaalla sadalla eurolla, ja levyaseman saattaa saada jopa ilmaiseksi. Mainittakoon, että Commodore 64:n diskettejä jäljentäessä on etua modatusta 5,25" levyasemasta, jolloin disketin molemmat puolet saadaan luetua yhdellä kertaa. Kaikki yleiset levykuvamuodot ovat tuettuja ja tällä hetkellä muun muassa Amigan ja Atarin IPF ja ADF-tiedostoja voi kirjoittaa takaisin levyille.

Kasettien tallennuksessa voidaan käyttää yksinkertaisimmillaan äänikorttia ja kasettinauhuria. Parhaasta päästä kasettien arkistointiin tarkoitettu laitteistosta on Commodore 64:lle huonosti saatavilla oleva DC2N. Tämän lisäksi hiljattain 1541 Ultimate II:een toteutettiin kasettiasema-adapteri, joka on firmware-versiosta 2.6 eteenpäin vartenotettava vaihtoehto kasettien tallentamiseen myös Turbo tape -nauhojen osalta.

Kasettien tallennuksessa voidaan käyttää yksinkertaisimmillaan äänikorttia ja kasettinauhuria. Parhaasta päästä

kasettien arkistointiin tarkoitettu laitteistosta on Commodore 64:lle huonosti saatavilla oleva DC2N. Tämän lisäksi hiljattain 1541 Ultimate II:een toteutettiin kasettiasema-adapteri, joka on firmware-versiosta 2.6 eteenpäin vartenotettava vaihtoehto kasettien tallentamiseen myös Turbo Tape -nauhojen osalta.

Avaamaton paketti on peliarkistoinnin kannalta ehdottomasti mielenkiintoisin kohde, mutta usein avaamattomassa paketissa olleiden levyjen pinnasta voi löytyä esimerkiksi orastavaa hometta. Käytettyjen diskettien pinnassa saattaa taas olla merkittävässä määrin likaa tai tahroja. Onneksi tähänkin on lääke: pullo isopropanolia ja pari pumpulipuikkoa. Disketin voi yrittää puhdistaa molemmin puolin sen lukupäälle tarkoitettu aukosta tai siirtää varsinaisen kiekon uuden disketin kuoriin. Jälkimmäisessä tapauksessa disketti leikataan kyljestä auki ja kiekko siirretään toiseen, varmasti puhtaaseen koteloon. Tähän tosin turvaudutaan harvoin. Levyn läpikäynti voi kestää noin 10–15 minuuttia, ja sen voi joutua tekemään useita kertoja. Sitkeys kuitenkin saatetaan palkita virheettömällä jäljennöksellä.

Missä mennään arkistoinnin suhteen?

Skenemateriaalia on tallessa varsin hyvin useimmille alustoille, mutta edelleen löytyy harvinaisuuksia ja timantteja. Pelien osalta tilanne on kohtalainen, mutta esimerkiksi Commodore 64:n osalta hyvin merkittävä osa on edelleen arkistoimatta, vaikka arkistointia tehdään jatkuvasti. Harvinaisempien alustojen osalta organisointi puuttuu tosin kokonaan, ja ne ovatkin tällä hetkellä pahimmassa vaarassa. Satunnaisia arkistioijia on paljon, mutta säännöllisesti arkistioivia on maailmallakin käytännössä vain vähän. Suurin osa jäljennöksistä tulee muutamilta aktiivisilta keräilijöiltä tai harrastajilta. Yhteistä aktiiveilla on se, että he ovat usein keski-ikäen kynnyksellä olevia miehiä ja ovat usein monessa mukana. Commodore 64:n kasettien ja diskettien osalta tehokkaaksi menetelmäksi on osoittautunut Jarkko Lehden vuosittain järjestämä Kasettilamerit-niminen tapahtuma, jossa jäljentämiseen käytetään porukalla koko viikonloppu. Viime vuonna tapahtumassa modattiin muutamia levyasemia jäljennyskäyttöön sekä arkistointiin pari kassillista Turbo Tape -nauhoja letkeissä tunnelmissa. Myös muutamia levykkeitä luettiin, mutta asemat saatiin käyttökuntoon vasta myöhemmin.





Nyrkkisääntönä voi todeta, että mitä harvinaisempi alusta on, sitä varmemmin sen tuotoksia ei ole vielä arkistoitu. Lisäksi diskettien arkistointi on tällä hetkellä kenties laajemmin organisoitua kuin kasettien. Kasettien arkistointia haittaa helpon ja nykyaikaisen arkistointilaitteen puute, koska monikaan ei halua viritellä kasettinauhuria äänikortin perään, eikä välttämättä tiedä miten toimivan jäljennöksen saa sieltä ulos.

Arkistoinnin ongelmia

Aika on tehnyt tehtävänsä, ja disketeistä ja kaseteista tullut paikoin hyvin harvinaisia. Varakkaat tai pelkästään peleihin huomionsa kohdistavat keräilijätäkään eivät helpota arkistointiharrastusta. Yllättävän monet kieltäytyvät osallistumasta arkistointiin materiaaliaan lainaamalla, vaikka laitteiden kanssa voidaan tarvittaessa jopa tulla paikan päälle. Kotimaisten tuotosten arkistoinnissa on juuri tämä ongelma. Niitä on vaikeaa löytää, ja löydettyä ei omistajalta välttämättä saa lupaa lukea levyä tai kasettia talteen, vaikka tarkoitus on saada peli arkistoitua niin, että alkuperäinen omistajakin voi nauttia siitä myös levykkeen tuhoutumisen jälkeen. Amersoftin peleistä on tähän mennessä arkistoitu vain yksi. Moni ei tunnu tiedostavan, että kalleinkin peli tai parhaiten säilytetty skenelevy tulee tuhoutumaan noin 30 vuoden jälkeen,

eikä levyä voi enää ladata tai palauttaa. Huolimatta siitä miten diskettia säilyttää, magneettisen median rajat tulevat ennen pitkää vastaan. Ei ole tavatonta, että ulkomaan markkinoilla saa pulittaa tavanomaisistakin peleistä kymmeniä dollareita, puhumattakaan harvinaisuuksista joiden hinnat menevät yli tuhannen dollarin.

SPS:llä on oma IPF-formaatti, jonka tarkoituksena on toimia pelin autenttisenä tiedostomuotona, ja jota voidaan käyttää emulaattoreiden avulla kuten alkuperäistä. Tällä hetkellä alkuperäisten pelien saatavuus IPF-muodossa on heikkoa. Netistä löytyy SPS:n vuotanut Amiga-peliarkisto, mutta se ei ole ollut ajan tasalla aikoihin. Syy IPF-tiedostojen huonoon saatavuuteen ovat tekijänoikeudet. Vanhojakaan pelejä ei saa pistää vapaasti jakoon, sillä on edelleen tahoja, jotka omistavat oikeudet, eivätkä halua luopua niistä. SPS tekee kuitenkin jatkuvasti töitä, jotta pelit voidaan julkaista tulevaisuudessa laillisesti, ja näin on joidenkin nimikkeiden osalta tehtykin. Periaatteessa ainoa tapa saada pelin IPF-tiedosto itselleen on ottaa SPS:ään yhteyttä ja esittää todisteet siitä, että omistaa kyseisen pelin alkuperäisenä.



Kasettilamerit — suomalainen arkistointitapahtuma

Kasettilameri on varmasti osalle lukijoista outo termi. Sen juuret ovat Philipsin 1960-luvulla kehittämässä C-kasetissa, joka oli pitkään yleisin musiikin tallennukseen käytetty media. 1980-luvulle saavuttaessa sitä käytettiin myös tietotekniikassa tiedon tallennusmedia. Sana lameri puolestaan viittasi aikanaan harrastajaan, jonka kyvyt tai tuotokset eivät yltäneet riittävän korkealle muun harrastajapiirin hieman elitistisestä näkökulmasta katsottuna. Levyaseman omistajien mielestä kasetilta ohjelmiaan lataavat olivat kasettilamereita. Tästä Kasettilamerit-tapahtuma sai nimensä.

Commodore 64:llä on jo pidemmän aikaa siirretty ohjelmia disketeiltä moderneihin formaatteihin ja arkistoitu erilaisiin tietokantoihin. Kasettien laajasta suosiosta huolimatta varsinkin Turbo Tape -muodossa nauhoitetut kasetit ovat kuitenkin jääneet täysin huomiotta. Vaikka samat kopiopelit on kopioitu lukuisille eri kaseteille täsmälleen samanlaisina kopion kopioina, kaseteilta löytyy aina silloin tällöin täysin uutta materiaalia.

Kasettilamerit tapahtumana keskittyi alun perin täysin kasettien arkistointiin ja vasta hieman myöhemmin laajeni diskettien puolelle yleiseksi Commodore

64-aiheiseksi arkistointitapahtumaksi. Tapahtuma on ollut pienen mutta intohimoisen C64-harrastajajoukon kokoontuminen, jonka jäsenet ovat muunmuassa ohjelmoineet omia työkaluohjelmia sekä muokanneet rautaa parempien jäljennösten aikaansaamiseksi. Tästä kehitystyöstä suuri kunnia kuuluu Jorma Oksalle ja Tommi Lempiselle.

Kaseteilta on tähän mennessä saatu tietoa talteen yli 75 megatavua. Yksittäisiä ohjelmia on yli 2800 kappaletta — määrä, joka täyttää Commodore 64:n muistin yli 1200 kertaa. Jäljennetty data on suurelta osin vielä tarkempaa läpikäyntiä vailla, mutta listaukset ja MD5-summat kasettien sisällöstä ovat tallessa ja listattuina. Kaikki aiemmin tuntematon skenemateriaali siirretään tulevaisuudessa CSDb:n tietokantaan. Mahdolliset uudet tai aiemmin kateissa olleet pelikräkit siirretään c64heaven.com-projektille, joka yrittää kerätä kaikki mahdolliset eri versiot C64-pelien kräkeistä.

Miten voin osallistua?

Eräs tulkinta elämän tarkoituksesta on tiedon kerääminen ja eteenpäin siirtäminen. Arkistointi on pohjimmiltaan juuri sitä. Nykyhetkestä katsoen on vaikeaa ellei mahdollontaa tietää, millainen tieto on

tulevaisuudessa tärkeää ja mielenkiintoista. Mikä tahansa lukijoiden hyllyistä löytyvä vanha kasetti- tai diskettipohjainen materiaali saattaa sisältää toistaiseksi tuntemattomia aarteita, jotka vain odottavat uutta elämäänsä arkistoinnin kautta. Artikkelin kirjoittajat ottavat kaikki kasetit ja disketit mielellään vastaan jäljentämistä ja projekteihin toimittamista varten. Erityisen kiinnostavia ovat Turbo Tapet, skenedisketit ja suomalaiset Commodore 64 -pelit. Kasetista ja disketistä saa aina halutessaan jäljennettyn version takaisin, ja samalla varmuuskopion. Arkistointi siis jatkuu!

Turbo tape -nauhojen lahjoitukset:
Jarkko Lehti, grue@iki.fi
Alkuperäiset disketit ja kasetit:
Aki Sivula, aki.sivula@iki.fi

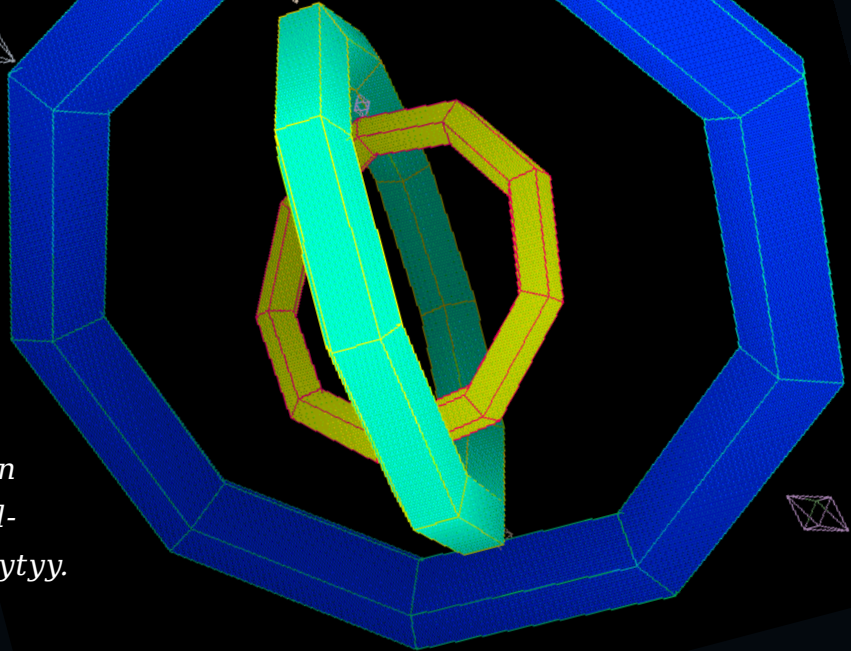


3D-rajapintojen lyhyt historia

Skrolli pureutuu tulevissa numeroissaan OpenGL-grafiikkaohjelmointiin. Sitä ennen on kuitenkin hyvä tehdä pikakatsaus reaaliaikaisen 3D-grafiikan taustoihin, sillä monikaan ei tule ajatelleeksi, mitä kaikkea tekniikan takaa löytyy.

Teksti: Mikko Rasa

Kuvat: Dave Fischer, Maarten Everts



Ensimmäinen askel kohti 3D-grafiikkaa otettiin jo renessanssin aikana, kun italialaiset kuvataiteilijat alkoivat käyttää maalauksissaan todenmukaista perspektiiviä. Piirtämällä rakennusten ääriviivoja peilin pinnalle Filippo Brunelleschi havaitsi viivojen jatkeiden kohtaavan horisontissa. Myöhemmin Leon Battista Alberti kirjoitti kirjan De Pictura, jossa hän loi matemaattisen perustan tasoprojektiolle. Piero della Francesca laajensi Albertin työtä omassa teoksessaan De Prospectiva Pingendi, joka kuvitettuna oli De Picturaa helpompi ymmärtää. Tasoprojektiota käytetään 3D-grafiikassa yhä tänäkin päivänä.

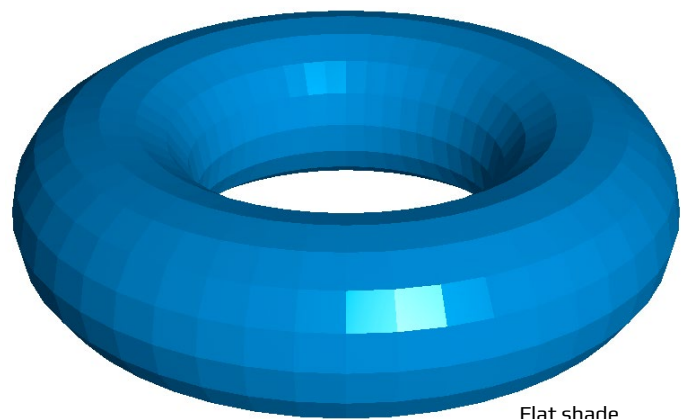
Reaaliaikaista 3D-tietokonegrafiikkaa nähtiin ensimmäisen kerran 1960-luvulla MIT:ssä, kun Ivan Sutherland kehitti ensimmäisen virtuaalikypärän. Laite oli erittäin kömpelö ja näkymä koostui yksinkertaisesta vektorigrafiikasta. Pian tutkimus siirtyi Utahin yliopistoon, jossa David Evans toimi vasta perustetun tietojenkäsittelytieteen laitoksen johtajana ja värvasi Sutherlandin alaisuuteensa.

Ennen yleiskäyttöisten rajapintojen aikakautta grafiikan piirtotutkimus toteutettiin kullekin sovellukselle erikseen, yleensä täysin ohjelmallisena pikselinpiirtona. Grafiikka koostui useimmiten viivoista tai tasavärisistä kolmioista, mutta etenkin peleissä käytettiin toisinaan myös muita, hyvinkin omalaatuisia lähestymistapoja.

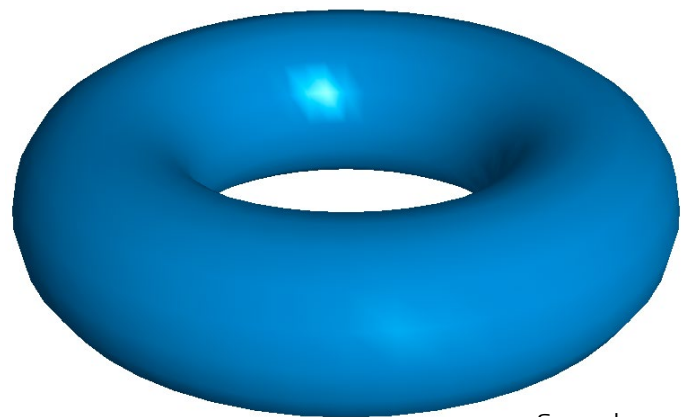
Ensimmäiset yleiskäyttöiset rajapinnat reaaliaikaiseen 3D-piirtoon julkaistiin vasta 80-luvun lopulla. Markkinoista kilpailivat avoimeen standardiin perustuva PHIGS (Programmer's Hierarchical Interactive Graphics System) ja Silicon Graphicsin IRIX-työasemilla käytetty IrisGL.

PHIGS:n toiminta perustui hierarkiseen esitykseen virtuaalimaailmasta ja siinä olevista esineistä. Tämä puurakenne sisälsi kaiken kuvan tuottamiseen tarvittavan tiedon. Esineen ominaisuuksiin tehdyt muutokset tulivat välittömästi näkyviin ruudulle. Tämä lähestymistapa teki rajapinnasta yksinkertaisen käyttää, sillä järjestelmä piti huolen kaikista laskentaan liittyvistä yksityiskohdista.

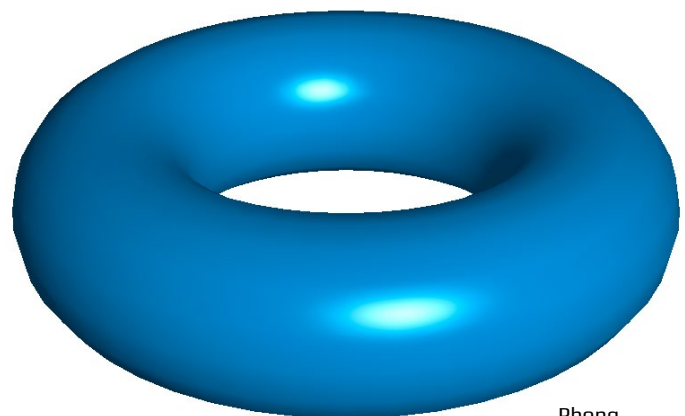
PHIGS ei tukenut lainkaan valaistusta eikä teksturointia. Vaikutelma valaistuksesta oli mahdollista luoda värityksellä mallin verteksit etukäteen laskettujen valaistusarvojen mukaan. Myöhemmin rajapinnasta julkaistiin uusi versio PHIGS+, johon oli lisätty reaaliaikainen valaistus ja kehittyneempiä grafiikkaprimitiivejä.



Flat shade



Gouraud



Phong

Alkuun näytti siltä, että PHIGSistä tulisi 90-luvun rajapinta laajemman tukensa vuoksi. Paineen alla SGI muokkasi myös omasta rajapinnastaan avoimen version ja julkaisi sen nimellä OpenGL. Ratkaisu osoittautui viisaaksi. OpenGL:n matalan tason lähestymistapa mahdollisti grafiikkakoodin paremman optimoinnin, mikä oli tuon ajan tietokoneilla erityisen tärkeää.

OpenGL:llä ei ole PHIGSin kaltaista pysyvää esitystä ruudun sisällöstä. Sen komennot suoritetaan välittömästi, ja ohjelman täytyy syöttää niihin liittyvä data joka kerta uudestaan. Tämä tekee ohjelmoinnista monimutkaisempaa, ja usein käytössä onkin joukko apufunktioita mallien ja niiden ominaisuuksien hallintaan. Matalamman tason rajapinta kuitenkin helpottaa toteutuksen tekijän työtä ja mahdollistaa uusien algoritmien nopeamman kokeilun ja omaksumisen.

Kiihdytetty 3D-grafiikka saapui kotikäyttäjien ulottuville aluksi pelikonsoleissa kuten Sonyn Playstationissa, mutta myöhemmin useat valmistajat julkaisivat 3D-ohjaimia myös pöytä-tietokoneille. Ensimmäiset PC:n ohjaimet tarjosivat ainoastaan perusprimitiivien piirron teksturoinnilla tai ilman, ja muu las-kenta piti tehdä pääsuorittimella.

OpenGL:n kanssa kilpaili 1990-luvun loppupuolen PC-maailmassa kaksi muuta rajapintaa. Microsoft oli vastikään julkaissut Direct3D:n Windows 95:n mukana, ja 3dfx:n Voodoo-kiihdyttimistä sai täyden tehon irti käyttämällä yhtiön omaa Glideä.

Sekä Glide että Direct3D perustuvat OpenGL:n tapaan välittömästi suoritettaviin piirto komentoihin. Ensi silmäyksellä rajapinnat vaikuttavat hyvinkin erilaisilta, mutta pintaa syvemmälle katsoessa selviää, että kyseessä on viime kädessä samat ominaisuudet eri kuorissa.

Glide oli alkuun vahvoilla. Yhteen piiriin sidottuna se tarjosi paremman suorituskyvyn kuin yhteensopivuutta painottaneet kilpailijansa. OpenGL toimi parhaiten työasemiin suunnatuilla kalliilla grafiikkakiihdyttimillä, kun taas Direct3D kärsi ajurien lastentaudeista ja kehittäjien epäluuloista uutta rajapintaa kohtaan.

Vuosituhanneen vaihteen lähestyessä niin kiihdytinkiirit kuin ajuritkin olivat kehittyneet, ja Glide menetti kilpailuetunsa. Viimeinen naula arkkuun oli, kun 3dfx:n huonot strategiset päätökset uuden piirisukupolven kehityksessä ajoivat yhtiön taloudellisiin vaikeuksiin ja lopulta konkurssiin. Voodoo-kiihdyttimet ja niiden mukana Glide-rajapinta katosivat markkinoilta.

2000-luvun alussa 3D-tietokonegrafiikka koki toistaiseksi suurimman mullistuksensa, kun kiihdytinkiirit muuttuivat ohjelmoitaviksi. Enää ei tarvinnut tyytyä kiinteisiin valaistumalleihin, vaan grafiikkaohjelmoija saattoi vapaasti määrittää haluamansa mallin. Myös monenlaiset erikoistehosteet tulivat mahdollisiksi. Ohjelmoitavat varjostimet olivat aluksi rinnakkainen vaihtoehto kiinteälle toiminnallisuudelle, mutta myöhemmin ne syrjäyttivät sen kokonaan.

2000-luvulla alkoi myös älypuhelin aikakausi, ja niistä kehittyi nopeasti minikokoisia tietokoneita. Ei kestänyt kauan, kun piirivalmistajat kehittivät niitä varten vähävirtaisia grafiikkakiihdytinkiirejä. Rajapintana näissä toimii OpenGL ES, joka on karsittu versio OpenGL:stä.

Uusimpiin 3D-rajapintoihin kuuluu selaimessa käytettävä WebGL, jolla 3D-sisältöä voidaan upottaa HTML-sivulle. Se perustuu OpenGL ES:ään, mutta Javascriptin rajoitusten takia jotkin asiat on toteutettu eri tavalla. Selaintuki on toistaiseksi vielä puutteellista, eikä tekniikkaa ole omaksuttu osaksi normaaleja nettisivuja.



Silicon Graphicsin Iris Indigo -työasema



GLX Gears -testiohjelma

Mitä klassikkokoneille kuuluu nykyään?

1980-luvulla tuli markkinoille kymmenittäin erilaisia mikrotietokoneita, joista useimpia ainakin joku muistelee lämmöllä. Muutamat näistä eivät kuitenkaan jääneet vain historian lehdille, vaan niiden ympärillä on edelleen harrastajavetoista ja kaupallistakin toimintaa. Skrolli esittelee kuusi klassikkokonetta, jotka eivät koskaan malttaneet kuolla pois. Ville-Matias Heikkilä Kuvat: Altera Corporation, Christian Wirth, Marcin Wichary, Risto Mäki-Petäys, Wikimedia Commons -käyttäjät Rama, Adam Kliczek ja EpiVictor

Yleisesti luullaan, että klassiset, alkuperäisten valmistajiensa jo aikoja sitten hylkäämät tietokoneet kelpaisivat korkeintaan vanhojen pelien pelailuun ja menneiden haikailuun. Tietokoneet ovat kuitenkin yleiskäyttöisiä laitteita, joiden mahdollisuuksia ei ole lyöty tehtaalla lukkoon. Vanhoillekin koneille voi siis ominaisuuksiensa puitteissa kehittää aivan uusia sovelluksia riippumatta siitä, mitä niillä aikoinaan tehtiin. Niille voi milloin tahansa tuoda uusia ohjelmia ja lisälaitteita, ja jos aitoa konetta ei enää saa, voi hankkia nykytekniikalla toteutetun uusversion.

Juttuun valitut koneet ovat kulttuuriltaan eloisia ja tarinoiltaan vaihtelevia. Ne eivät suinkaan ole ainoat klassikkokoneet, joissa on edelleen eloa – kymmenille muillekin laitteille on ainakin muutamia tuoreehkoja demoja ja pelejä, ja osa on saanut suosiota myös esimerkiksi soittimina.

Commodore 64

Commodore 64 tuli markkinoille vuonna 1982 ja saavutti johtoaseman useimmissa maissa, joissa sitä myytiin. Etenkin pohjoismaissa C-64 oli niin vahva kilpailijoihinsa nähden, että siitä muodostui oikea aikakautensa symboli monille 1980-luvulla lapsuttaan viettäneille.

Commodore valmisti ennen vuoden 1994 konkurssiaan noin 15 miljoonaa Kuusnelosta, ja se oli myös maailman myydyin tietokonemalli aina 2000-luvulle asti. Nykyisin esimerkiksi muutamia iPad-malleja on myyty enemmän.

Vaikka kaupallinen C-64-ohjelmien tuotanto olikin hiipunut ennen konkurssia, koneen asema tietyissä harrastajapiireissä oli jo tällöin hyvin vakaa. C-64-demojen ja SID-musiikin tuotanto on jatkunut katkeamattomana perinteenä nykypäiviin asti, ja C-64:lle on julkaistu enemmän demoja kuin millekään muulle alustalle. Demoskene on myös vienyt C-64-ohjelmoin-

titeknikaan sellaiselle tasolle, josta ei 1980-luvulla edes uneksittu.

Myös muut kasibittiset Commodoret ovat hyötynet C-64-yhteisön vahvuudesta. Commodore 16 -yhteensopiville koneille, joita ovat myös C-116, C-264 ja Plus/4, on ollut jonkin verran demotuotantoa 1980-luvulta asti. 2000-luvun puolella muutamat ryhmät innostuivat myös VIC-20:stä, jonka demoille on ominaista on pitäytyminen laajentamattoman koneen 5 kilotavun muistissa.

Commodoren konkurssin jälkeen on ilmestynyt myös uusia C-64-lisälaitteita. Käyttökelpoisin C-64-käyttäjälle lienee hollantilaisen Gideon Zweijterin 1541 Ultimate -moduli, joka simuloi 1541-levyasemaa täydellisesti mutta käyttää lerppujen sijaan SD-korttia. Moduli toteuttaa myös 16 megatavun REU-muistilaajenuksen, ja jotkin mallit toimivat myös verkkokortteina. CMD-yhtiön SuperCPU-laajennos taas sisältää 20 megahertsin taajuudelle kellotetun 16-bittisen 65816-suorittimen, joka korvaa koneen oman 1-megahertsisen 6510:n, ja jolle on toteutettu mm. reaaliaikaiset MP3- ja MOD-tiedostojen soitinohjelmat.

Ehkä ylistetyin C-64:n komponentti on SID-äänipiiri, jota kaivattaisiin mm. erilaisiin syntetisaattoriprojekteihin enemmän kuin koko maailmasta löytyy. Vuonna 1997 julkistetun Sidstation-syntetisaattorin valmistaja Elektron haali kuuleman mukaan kaikki varastoihin jääneet piirit, ja siltikin laitteita jouduttiin lopulta myymään tyhjiillä piirikannoilla. Monet ovat yrittäneet luoda SIDiä uusiksi, mutta kukaan ei ole vielä uudelleentoteuttanut sen analogisia suotimia tyydyttävästi.

Amerikkalainen Jeri Ellsworth on uudelleentoteuttanut koko C-64:n kahdessakin eri projektissaan, C-Onessa ja ulkoisesti peliohjainta muistuttavassa DTV:ssä. Molemmissa on myös uusia grafiikkatiloja. On myös huhuttu, että 1541 Ulti-





siin vuonna 1994, Amiga oli jo jäänyt monin tavoin jälkeen kilpailijalleen PC:lle. PC oli huomommin suunniteltu mutta pystyi etenkin 486-suorittinten yleistyttyä paikkaamaan puutteensa raa'alla laskentateholla. Amiga 4000:ssa ja 1200:ssa käytetty AGA-piirisarja ei parantanut grafiikkaominaisuuksia riittävästi, eikä PC:n päihittäjäksi lupailtu uusi grafiikkarauta valmistunut koskaan. Konkurssinjälkeisistä Amiga-laitteista tärkeimpiä olivatkin saksalaisen Phase5:n prosessorikortit ja näytönohjaimet, joilla suorituskyky saatiin pidettyä PC:n tasolla.

Kun PC-rauta meni ohi myös turbovarustelluista Amigoista, tuumattiin Amiga-yhteisössä, että ohjelmistotason eleganssissa Amiga päihittää kilpailijansa edelleen. Niinpä kehitettiin täysin uusia tietokoneita, jotka ajavat AmigaOS-johdannaisia mutta eivät ole yhteensopivia klassisen laitearkkitehtuurin kanssa. Huomattavimpia näistä ovat vuosina 2000-2006 valmistettu Pegasos ja vuodesta 2002 tuotetut AmigaOne-koneet. AmigaOne-sarjan uusin on vuonna 2010 julkistettu X1000, jonka suorittimena on 64-bittinen PWRficient-tuplaydinsuoritin.

Amigan aineettomia oikeuksia on suojeltu Commodoren konkurssin jälkeen hyvin mustasukkaisesti, mikä on jakanut kehittäjäyhteisön. Yhdessä leirissä ovat Amigan tavaramerkin omistava amerikkalainen Amiga Incorporated ja sen luvalla AmigaOS:n ja AmigaOnen kehityksestä vastaava Hyperion Entertainment. Kapinallisleiriä edustavat puolestaan MorphOS- ja AROS-käyttöjärjestelmät ja niitä ajavat koneet. MorphOS ja uudet AmigaOS:t toimivat vain PowerPC-koneissa, muutamat Macintosh-mallit mukaanlukien, kun taas avoimen lähdekoodin AROS toimii myös X86-PC:issä ja jotenkuten jopa 68K-Amigoissa.

Monien amigistien mielestä pelkkä osittainen ohjelmistoyhteensopivuus ei tee koneesta Amigaa, vaan tarvitaan perinteisiä laitteita tai niille uskollista uustuotantoa. Vuonna 2007 avoimena lähdekoodina ja kytKentäkaaviona julkaistu Minimig toteuttaa Amiga 500:n koko logiikan yhdellä FPGA-piirillä, ja se on saatavilla paitsi valmiina emolevynä, myös C-Onen logiik-

katiedostona. NatAmi taas on pitempään kehitelty 68060-pohjainen AGA-Amiga, joka pyrkii täydelliseen laiteyhteensopivuuteen ilman väylälätensseja ja muita vanhan Amigan rajoitteita.

Amiga on C-64:n ohella merkittävin demotaidetta määrittänyt laitealusta, ja Amiga-demojen tuotanto jatkuu katkeamattomana perinteenä. Demot ovat kuuluneet alusta asti myös valtavirtaisempaan Amiga-kulttuuriin, joskin monia uudempien Amigoiden nimeen vannovia kismittääkin demontekijöiden vastahakoisuus tukea uudempaa tekniikkaa. Valtaosa demoista on suoritinkeskeisiä AGA-demoja, mutta myös 1980-luvun Amigoille tehdään demoja.

Acorn Archimedes

Brittiläinen Acorn lienee tämän jutun nimistä suomalaislukijalle vierain. Acornin vuonna 1987 julkaisema Archimedes kilpaili Amigaa ja Atari ST:tä vastaan mutta ei muodostunut hitiksi nopeudestaan huolimatta. Archimedestä varten kehitetty ARM-suoritinarkkitehtuuri jäi kuitenkin elämään. ARM tarjosi paljon tehoa pienellä logiikkamäärällä, ja se on tällä hetkellä kännykkäteollisuuden ansiosta maailman yleisin suoritinarkkitehtuuri.

Vaikka Archimedes jäikin meillä oudoksi marginaalikooneeksi, se sai kotimaassaan uskollisen käyttäjäkunnan. Acorn lanseerasi ensimmäisen Archimedeen jälkeen useita muita malleja, joista se siirtyi vuonna 1994 RiscPC-sarjaan. Vuonna 1998 Acorn luovutti RiscPc:n valmistusoikeudet Castle Technologylle, joka jatkaa RISC OS -käyttöjärjestelmän kehittämistä tänäkin päivänä.

Acornin koneiden nykyperillinen on myös Raspberry Pi, jonka nimekkäin taustamies on Elite-pelistä tunnettu ja myös Archimedeelelle pelejä ohjelmoinut David Braben. Pi'lle on olemassa virallinen versio RISC OS:stä, ja myös huomattava osa Archimedeen peleistä toimii myös Pi:ssä sellaisenaan.



Nintendo Famicom/NES

Famicom tuli Japanin markkinoille vuonna 1983 ja pari vuotta myöhemmin muualle maailmaan nimellä Nintendo Entertainment System. Niissä maissa, joissa NESiä ei virallisesti myyty, kone peleineen tuli tutuksi lukemattomina piraattikopioina.

Taiwanilainen puolijohdevalmistaja UMC kehitti 1980-luvun lopulla kopiot Famicomin prosessorista ja videopiiristä. Näin monet Itä-Aasian elektroniikkavalmistajat pääsivät tuottamaan omia Famicom-kloonejaan. Myöhemmin UMC ja muutamat muutkin alkoivat valmistaa myös NOACEja eli yhden piirin Famicom-toteutuksia (NES-On-A-Chip). Edistyneimmissä, kuten V.R.Technology-yhtiön VT03:ssa, on alkuperäiskonetta laajemmat grafiikka- ja ääniominaisuudet.

Piraattituotanto on jatkunut keskeytyksettä näihin päiviin asti. Kloonien hinnat ovat tuotantomääristä johtuen niin alhaalla, että köyhien maiden alemmilla keskiluokillakin on varaa niihin. Mielenkiintoisia ovat etenkin "opetustietokoneet" kuten Gold Leopard King ja Asder PC, jotka ovat näppäimistökuorien sisään rakennettuja Famicom-klooneja. Näiden mukana tulee yleensä BASIC-tulkki, opetusohjelmia ja yksinkertaisia tekstinkäsittely-, piirto- ja laskentaohjelmia. Vuonna 2008 perustettu amerikkalainen Playpower-säätiö tuottaa etenkin kehitysmaihin tarkoitettuja opetusohjelmia ja -pelejä näille äärimmäisen halvoille tietokoneille.

Vaikka viimeinen Nintendon hyväksymä Famicom-peli julkaistiinkin 1994, yhtiö tuki tuotetta aina vuoteen 2003 asti. Tällöin tosin myös Nintendon patentit raukesivat, ja lailliset Famicom-kloonit mahdollistuivat. Tunnetuin näistä lienee amerikkalaisen Messiah Entertainmentin Generation NEX.

Vuonna 1999 perustettu kiinalainen Nice Code Software on tunnetuimpia luvattomien kaupallisten Famicom-pelien kehittäjiä. Myös harrastajavetoinen ohjelmistotuotanto keskittyy lähiinnä peleihin, mutta etenkin eurooppalaiselle NESille on tehty myös jonkin verran demoja.



IBM PC

Alkuperäinen IBM PC tuli markkinoille elokuussa 1981. Siinä oli 4,77 megahertsin 8088-suoritin, 64 kilotavun RAM-muisti ja 320x200 pikselin nelivägrafikka. Massamuistiksi oli tarjolla kasettiasema ja 5¼ tuuman levykeasema, ja näytöksi riitti tavallinen TV-vastaanotin. PC:n ROM-muistissa oli valmiina Microsoftin BASIC-tulkki, joka käynnistyi, jos DOS-levyke ei ollut käynnistyksen aikana asemassa.

PC:n piti alkujaan kilpailla Apple II:n kaltaisten 8-bittisten mikrojen kanssa, mutta sen hinta oli turhan korkea ja monet ominaisuudet keskinkertaiset. PC:n pelasti kuitenkin IBM:n uskottava nimi, joka toi sille suosiota ammattikäytössä. Parin vuoden päästä IBM julkistikin ammattikäyttöön tarkoitettut XT- ja AT-mallit, joissa oli jopa kiintolevy vakiovarusteena.

Valtaosa nykyajan läppäreistä ja työpöytäkoneista on laitetasolla IBM PC-yhteensopivia. PC:t käynnistyvät edelleen 8088-yhteensopivassa "reaalitilassa", josta voi käynnistää nykykäyttöjärjestelmän sijaan vaikkapa muinaisen MS-DOS-version. Sen komentoriviltä voi puolestaan käynnistää esimerkiksi Alley

Cat -pelin, joka toimii suoraan, koska uudetkin näytönohjaimet ovat VGA-yhteensopivan osajärjestelmänsä kautta CGA-yhteensopivia.

PC onnistui menestymään teknisestä keskinkertaisuudestaan huolimatta. Se jopa syrjäytti käytännössä kaikki kilpailijansa: Applen Macintoshitkin ovat nykyään IBM PC -yhteensopivia. Menestykselle voi löytää parikin syytä. Ensinnäkin IBM:n uskottavuus toi PC:lle niin paljon toimistokäyttäjää, että siitä tuli toimistotekniikan ja myöhemmin muidenkin alojen de-facto-standardi. Toiseksi IBM:n moka PC-arkkitehtuurin lisensoinnissa mahdollisti halpojen PC-kloonien vyöryn markkinoille.

PC:tä on ollut hyvin vaikea uudistaa sen jälkeen, kun IBM menetti asemansa useiden laitevalmistajien sekamelskalle. Esimerkiksi jo syntyessään konservatiivista X86-käskykantaan on yritetty korvata paremmalla noin kymmenen vuoden välein, mutta niin DEC:n Alpha kuin Intelin Itaniumkaan eivät menestyneet valtavirtakoneissa. Viime aikoina on puhuttu ARM-arkkitehtuurista X86:n syrjäyttäjänä, ja jää nähtäväksi, onnistuuko siirtymä tällä kertaa. Ainakaan X86-koodin suoritusnopeus ei ole enää useimmille

PC-käyttäjille niin tärkeä kriteeri kuin 10 tai 20 vuotta sitten.

Pitäisikö X86:n ja ehkä koko PC-arkkitehtuurin syrjäytymistä odottaa innolla vai kauhulla? Toisaalta laiteyhteensopivuuden riippakiven häviäminen antaa tilaa tekniselle kehitykselle, mutta toisaalta laitealustan yleisyys ja avoimuus estää valtaa karkaamasta käyttäjiltä valmistajien käsiin. PC-arkkitehtuurin syrjäydyttyä ei välttämättä voi enää ostaa maitokaupasta Windows-läppäriä ja asentaa siihen Linuxia, vaan moinen rikollisuus vaatii samantapaista suojausten murtamista ja mallikohtaista erityissääntöä kuin mobiili- ja konsolimaailmassa on vaadittu jo pitkään.

Vaikka PC ja X86 katoaisivatkin valtavirran tietotekniikasta tämän vuosikymmenen aikana, tulevat ne varmasti elämään vielä kymmeniä vuosia harrastajapiireissä samoin kuin muutkin klassiset laitealustat. Nimekkäät muotivimpaimet kehittyvät vuosien mittaan aina vain suljetummiksi, monimutkaisemmiksi ja arvaamattommiksi, mutta klassiset alustat pysyvät yhtä hallittavina, tuttuina ja turvallisina kuin ennenkin.

Mikä on FPGA?

Klassikkokoneiden uusversiot on hyvin usein toteutettu FPGA-tekniikalla, joka kuuluu ohjelmoitavaan logiikkaan.

Perinteisten mikropiirien toiminta on lyöty tehtaalla lukkoon, mutta ohjelmoitavien logiikkapiirien rakenne on avoin. Niissä on ikään kuin kasa rakennuspalikoita, joista voi muodostaa kaikenlaisia digitaalisia piirejä. FPGA-piirit lataavat logiikkansa ulkoisesta muistista, joten ne voidaan muuttaa aivan toisenlaisiksi pelkästään vaihtamalla logiikkatiedostoa.

Nykyaikaisissa FPGA-piireissä on sen verran tilaa, että yhdelle sellaiselle mahtuu minkä tahansa kasibittisen koneen koko logiikka suorittimeen, erikoispiireineen ja muisteineen. Uuskotimikrot on huomattavasti kevyempi suunnitella FPGA:lla kuin perinteisillä mikropiireillä, eikä tekniikalle ominainen tila- ja energiasyöppöys juuri haittaa tässä sovelluskohteessa.

Tunnetuin FPGA-pohjainen uuskotimikro lienee amerikkalaisen Jeri Ellsworthin C-One. Se on tarkoitettu etenkin Commodore 64:n uustoteutukseksi, mutta logiikkatiedostoa vaihtamalla se muuntautuu moneksi muuksikin. Muita C-Onen jäljittelemiä koneita ovat VIC-20, Amiga 500 (Minimig), Amstrad CPC ja ZX Spectrum 48. Toki koneeseen voi ladata myös täysin omia logiikkaytimiä – esimerkiksi ruotsalainen Linus "lft" Åkesson on toteuttanut Parallelogram-demoaan varten aivan oman laitealustansa, joka toimii C-Onen toisen FPGA:n sisällä.

Omia emolevyjään käyttäviä uuskotimikroja ovat 1chipMSX (MSX-2), ZX Evolution (ZX Spectrum), Experiment-S Suska (Atari ST) ja Amiga-toteutukset Minimig ja NatAmi. Näistä ZX Evolution ja NatAmi eivät toteuta koko konetta FPGA:lla, vaan niissä on aidot Z80- ja 68060-suorittimet.

FPGA-kone saattaa äkkiseltään vaikuttaa pelkältä emulaattoria ajavalta pikkukoneelta, mutta eroja on esimerkiksi ajoituksen tarkkuudessa. Kun aidon tai FPGA-toteutetun C-64:n grafiikan taustaväriä vaihtaa, videosignaali muuttuu samalla mikrosekunnilla, mutta emulaattorilla muutos näkyy vasta seuraavalla näytönvirkistyksellä. Grafiikassa tämän viiveen vielä kestää, mutta esimerkiksi lisälaitteet ovat usein nirsoja ajoitusvaatimuksiltaan. FPGA-koneita käyttäessä on myös varmaa, ettei emulaattoria ajava käyttäjärjestelmä tule väliin aiheuttamaan nykimistä.



Toistaiseksi klassikkokoneiden ohjelmat toimivat emulaattoreilla paremmin kuin FPGA-koneilla, mutta asia saattaa muuttua lähivuosina. Logiikankuvauskielillä on sen verran luontevampi kuvata mik-

ropiirien toimintaa kuin tavallisilla ohjelmointikielillä, että jossain vaiheessa emulaattorikehittäjät voivat siirtyä käyttämään samoja VHDL- ja Verilog-lähdekoodeja kuin FPGA-kehittäjätkin.

QML – deklarattiivisen käyttöliittymän alkeet

QML (Qt Meta Language) on Digian Qt-ympäristöön kuuluva käyttöliittymien toteutukseen tarkoitettu kuvauskieli, jolla voi tehdä näyttäviä käyttöliittymiä jopa ilman ohjelmointitaitoa.

Ville Ranki

QML julkaistiin vuonna 2009, ja se on Nokian Harmattan-alustan ja MeeGon ”viralinen” tapa toteuttaa käyttöliittymiä. Näkyvin QML:n käyttäjä lienee tällä hetkellä Nokian N9-puhelin. Tulevaisuudessa ainakin Jollan Sailfish ja Ubuntu Mobile käyttävät QML:ää käyttöliittymissään.

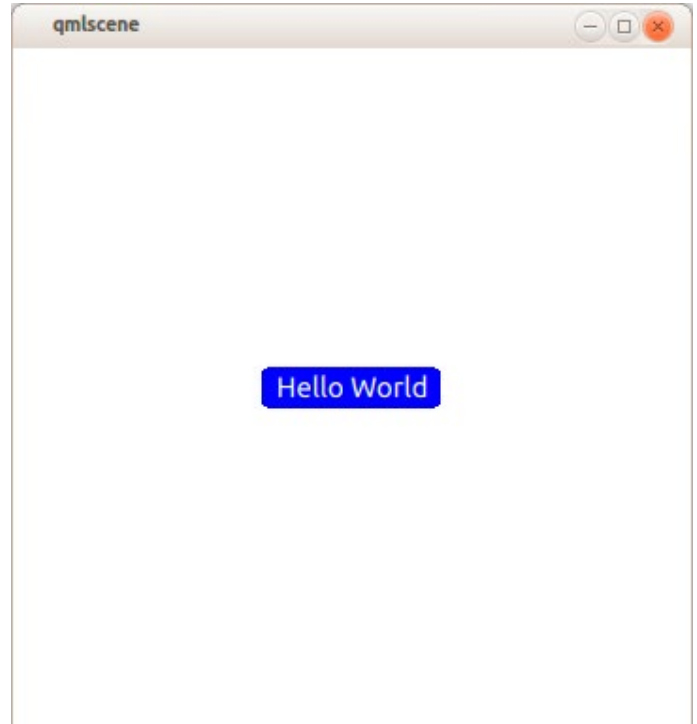
Perinteisesti graafiset käyttöliittymät on rakennettu sovellusohjelmointikielellä käyttöliittymäkomponenteista eli widgeteistä. Tämä on hidasta ja virheeltä. Toki tarkoitukseen on suunnittelu-sovelluksia, mutta niillä on omat rajoitteensa. Etenkään jos halutaan tehdä helposti ja nopeasti samantapaisia animoituja käyttöliittymiä kuin vaikkapa iPhonessa, perinteiset lähestymistavat eivät toimi.

QML eriyttää käyttöliittymän kuvauksen kokonaan sovellusohjelmointikielestä eli tyypillisesti C++:sta.

Qt-kirjasto tekee kuitenkin QML:n ja C++:n yhteistyön hyvin helpoksi. Kun QML:n rajat tulevat vastaan, voidaan toiminnallisuutta helposti laajentaa niin C++:lla kuin webbiohjelmoinnista tutulla Java-Scriptilläkin. Tyypillinen QML:ää käyttävä sovellus on C++-ohjelma, joka vain avaa QML:llä toteutetun käyttöliittymän.

QML:ää kannattaa käyttää kosketusnäyttö- tai hiirive-toisissa sovelluksissa, jotka pyörivät omassa ikkunassaan tai koko ruudulla. Mobiilimaailman ulkopuolella tällaisia voisivat olla XBMC:n kaltaiset mediatoistimet tai Angry Birdsin kaltaiset 2D-pelit. Peleissä fysiikkamotorin voisi toteuttaa C++:lla ja peligrafikat QML:llä. Perinteiseen ohjelmointiin kannattaa turvautua, jos sovellus käyttää useampaa ikkunaa, esittää paljon dataa tai sisältää paljon syöttöomakkeita. QML:ää voi kuitenkin käyttää myös osana widget-pohjaista sovellusta.

Kuten koko Qt, myös QML on tarkoitettu alustariippumattomaksi. Saman QML-sovelluksen saa tällä hetkellä toimimaan suoraan Windowsissa, työpöytä-Linuxeissa, OS X:ssä, Maemossa, MeeGossa, Symbianissa ja Windows CE:ssä. Android-QML toimii myös, vaikka onkin vielä kehityksasteella. iOS- ja WP8-portaukset ovat tekeillä.



Esimerkki 1: Ensitervehdys maailmalle

QML-ohjelmointi on helpointa aloittaa asentamalla Digian Qt SDK. Qt Creator, virallinen Qt-kehitysyökalu, tukee täysin QML-ohjelmointia. Näissä esimerkeissä on käytetty Qt SDK:n versiota 5.0 ja sen mukana tulevaa Qt Creator 2.6.1:tä. Käynnistä Qt Creator, valitse alkuruudussa ”Create

Project”, ”Applications” / ”Qt Quick 2 UI” ja anna projektin nimeksi ”skrolli”. Qt Creator luo QML-tiedoston skrolli.qml, jonka sisältö on seuraava (Listaus 1).

Voit käynnistää ohjelman painamalla Ctrl-R. Tämä näyttää ikkunan, jossa lukee ”Hello World”. Ohjelma sulkeutuu klikkaamalla ikkunan sisällä.

```
import QtQuick 2.0

Rectangle {
    width: 360
    height: 360
    Text {
        anchors.centerIn: parent
        text: "Hello World"
    }
    MouseArea {
        anchors.fill: parent
        onClicked: {
            Qt.quit();
        }
    }
}
```

Listaus 1

Deklaratiivinen kuvaus	Perinteinen ohjelmointi
<pre>Image { width: 300 height: 200 source: "skrolli.png" }</pre>	<pre>Image *image = new Image(); image->setWidth(300); image->setHeight(200); image->setSource("skrolli.png");</pre>

Esimerkki kuvan luomisesta deklarattiivisesti ja perinteisellä widget-ohjelmoinnilla.

Ensimmäinen import-rivi lataa QML:n peruskomponenttikirjaston. Muut tarjolla olevat kirjastot riippuvat Qt:n versiosta ja ympäristöstä. Niitä voi ohjelmoida myös itse.

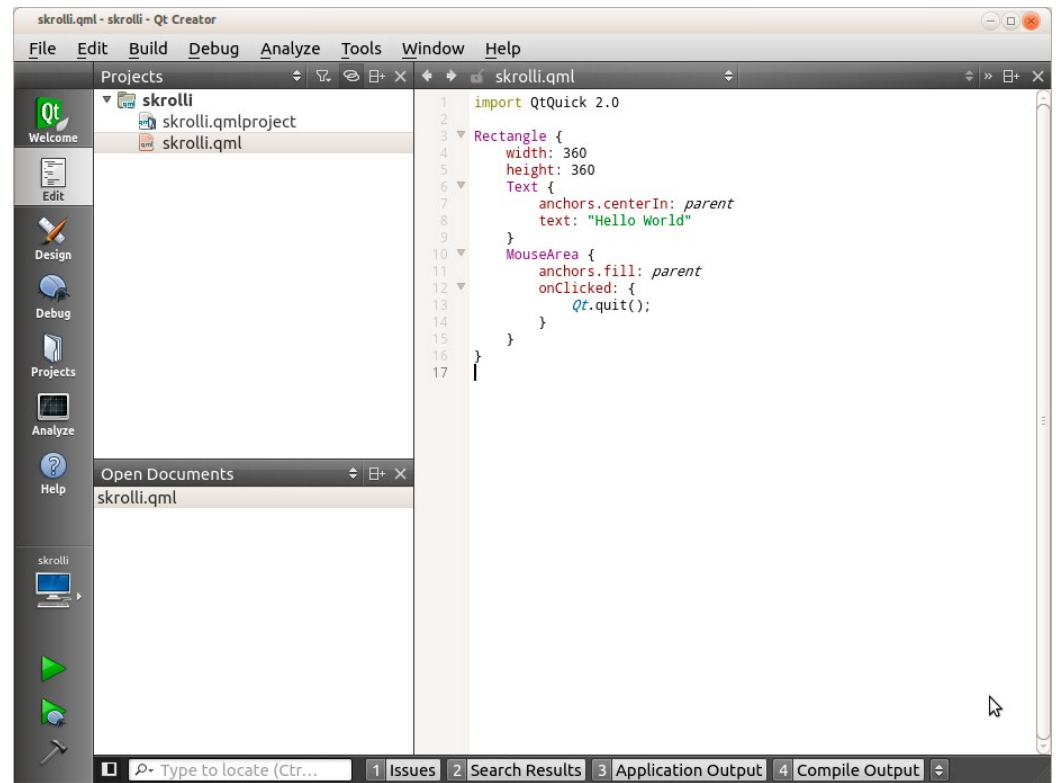
QML-dokumentti koostuu aaltosuluilla merkitystä lohkoista, joita kutsutaan elementeiksi. Ennen sulkeita annetaan elementin tyyppi. Sulkeiden sisällä määritellään elementin ominaisuudet (properties) ja lapsielementit. Esimerkissä kuvataan Rectangle-elementti eli nelikulmio, jonka koko on 360x360 pikseliä, ja jonka lapsina ovat Text- ja MouseArea-elementit.

anchors-ominaisuus määrittelee piirrettävän elementin niin kutsutut ankkuripisteet. Text-elementin *anchors.centerIn: parent* keskittää tekstin isäelementtinsä eli tässä tapauksessa nelikulmion keskelle. Tämä määrittely ei vaikuta Text-elementin kokoon, joten se on automaattisesti piirrettävän tekstin kokoinen.

Ankkuripisteillä voidaan vapaasti määritellä elementtien sijainnit. Esimerkiksi *anchors.left: toinenelementti*. *right* määrää elementin vasemman laidan olevan aina toisen elementin oikeassa laidassa. Ankkuripisteillä elementit pitävät paikkansa, vaikka niiden suhteelliset koot ja sijainnit muuttuisivat. Koot ja paikat kannattaa määritellä suhteessa toisiinsa aina kun mahdollista, jotta käyttöliittymä skaalautuisi erikokoisille näytöille.

MouseArea-elementti ei näy ruudulla, vaan sitä käytetään hiiren tai sormen klikkausten tunnistamiseen. Sen kooksi on määritelty *anchors.fill: parent*, joka asettaa sen koon ja paikan vastaamaan isäelementtiään eli nelikulmiota. MouseAreaan *onClicked* on signaalinkäsittelijä (signal handler) ja kutsuu JavaScriptillä ohjelman sulkevaa funktiota *Qt.quit()*.

Kaikista QML-elementeistä ja niiden ominaisuuksista saa Qt Creatorissa ohjeet



F1-näppäimellä. Ctrl-Space täydentää osittain kirjoitetun elementin tai ominaisuuden nimen, ja automaattinen sisennys onnistuu painamalla Ctrl-a ja Ctrl-i. Kommentteja kirjoitetaan kuten C++:ssa, eli esimerkiksi // muuttaa loppurivin kommentiksi.

Oman elementtityypin luominen

QML:ssä voidaan luoda omia elementtityyppejä yksinkertaisesti tekemällä uusi QML-tiedosto. Seuraavassa esimerkissä luomme MyButton-nimisen elementin.

Paina Qt Creatorin Projects-listassa oikealla napilla projektisi nimeä (skrolli) ja valitse "Add New". Aukeavasta dialogista valitse "Qt", "QML File (Qt Quick 2)". Anna tiedostolle nimi "MyButton". Naputtele tiedostoon Listaus 2.

Otamme nyt tämän elementin käyttöön skrolli.qml:ssä. Muokkaa sen sisältö vastaamaan listauksista 3 ja käynnistä painamalla Ctrl-r.

Omien elementtien määrittelyllä vältetään tiedostojen kasvaminen ylisuuriksi ja mahdollistetaan samojen elementtien uudelleenkäyttö.

```
import QtQuick 2.0
```

```
Rectangle {
    id: mybutton
    property string text: "Nappi"
    signal clicked
    width: textItem.width + 15
    height: textItem.height + 5
    color: "blue"
    radius: 5
    Text {
        id: textItem
        anchors.centerIn: mybutton
        text: mybutton.text
        color: "white"
    }
    MouseArea {
        anchors.fill: mybutton
        onClicked: mybutton.clicked()
    }
}
```

Listaus 2: MyButton.qml

```
import QtQuick 2.0
```

```
Rectangle {
    width: 360
    height: 360
    MyButton {
        anchors.centerIn: parent
        text: "Hello World"
        onClicked: {
            Qt.quit();
        }
    }
}
```

Listaus 3: skrolli.qml

MyButtonin juurielementti on nelikulmio (Rectangle), jolle on määritelty koon, värin ja reunojen pyöristyksen (radius) lisäksi id *mybutton*, ominaisuus *text* ja signaali *clicked*. Id on yksittäisen elementin tunniste, johon voidaan viitata sen lapsi- ja sisärelementeissä. Nelikulmion koko on esimerkissä määri-

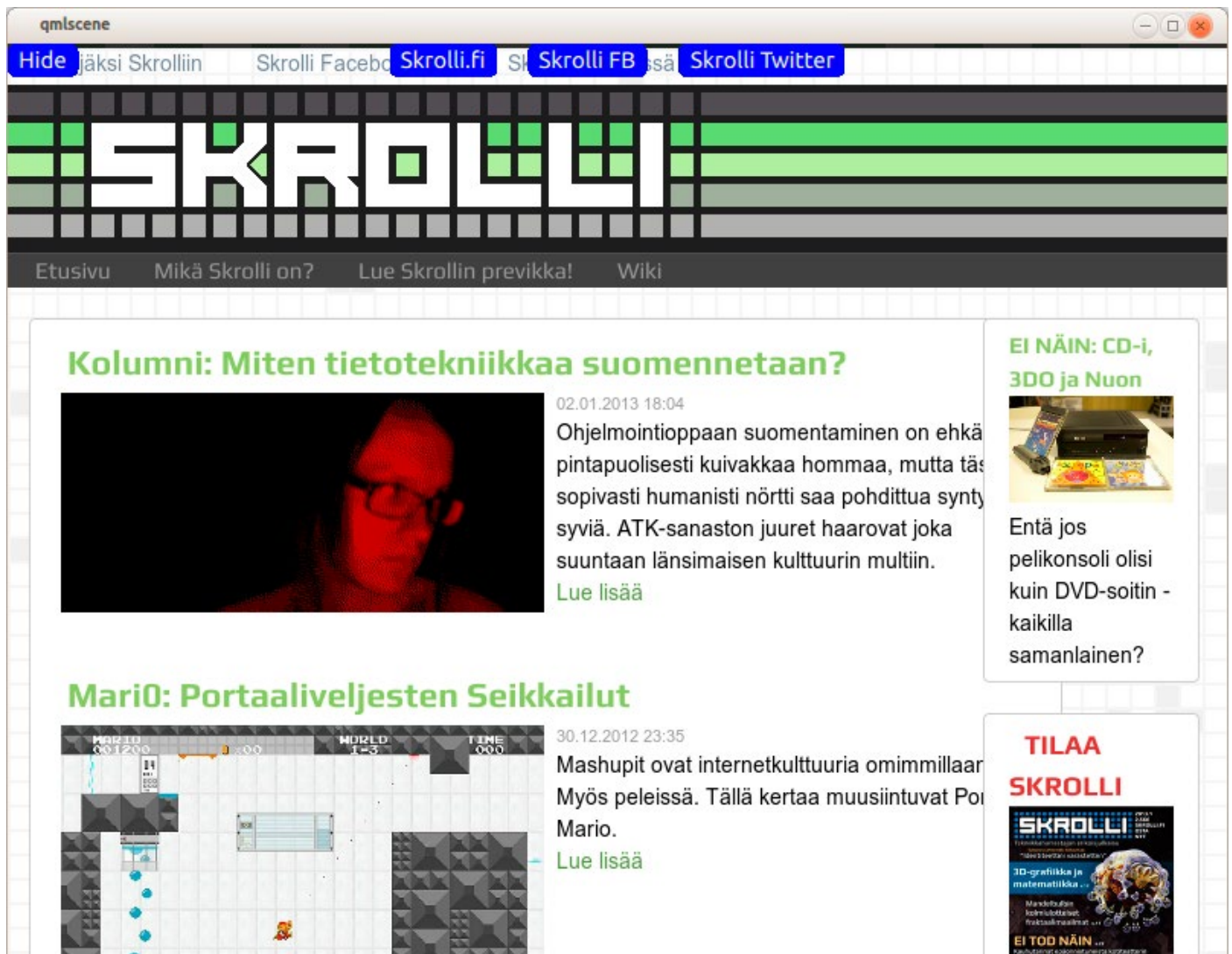
teltu suhteessa sen sisällä olevaan tekstielementtiin, jonka id on *textItem*.

QML-elementeille voidaan määritellä omia ominaisuuksia. MyButton.qml:n property-alkuinen rivi määrittelee tyyppille ominaisuuden *text*, joka on merkkijono (string) ja oletusarvoltaan "Nappi". Ominaisuutta käytetään Text-

elementissä ja sille asetetaan *skrolli.qml*:ssä uusi arvo "Hello World".

Napin painaminen aikaansaa *clicked*-signaalin sen sisällä olevaan *MouseArea*-elementtiin. Jotta tästä olisi hyötyä nappia käyttävissä ohjelmissa, on myös napin juurielementillä oltava oma *clicked*-signaali. *MouseArea*-

elementin signaalinkäsittelijä välittää *clicked*-signaalin eteenpäin MyButtonille, jonka signaalinkäsittelijä *onClicked* on määritelty *skrolli.qml*:n puolella ohjelman lopetuksiksi. Signaaleilla voi olla myös parametreja funktiokutsujen tapaan.



Esimerkki 2: Pieni verkkoselain

Seuraavaksi teemme yksinkertaisen www-selaimen käyttäen QML:n *WebView*-elementtiä. Yläreunaan tehdään ohjauspalkki, josta voidaan hiirellä siirtyä johonkin etukäteen annetuista osoitteista. Ohjauspalkin voi myös piilottaa ja palauttaa näkyviin.

Muokkaa *skrolli.qml* listaus 4:n mukaiseksi.

Juurena on yleiselementti *Item*, jonka täyttävät seuraavaksi tehtävä *WebTools*-elementti ja *QtWebKit*-kirjastosta tuleva *WebView*-elementti. *WebTools*in positiivinen z-koordinaatti varmistaa, että se piirretään aina *WebView*in päälle. Tiedostoon

WebTools.qml kirjoitetaan listaus 5.

Tiloilla (*states*) voi rakentaa käyttöliittymille mukavasti toimintalogiikkaa, esimerkiksi elementtien piilotus tarpeen mukaan. *WebTools*in juurena on *Item*, jonka tilat ovat "HIDDEN" ja "SHOWN". *PropertyChanges*-elementillä määritellään minkä ele-

mentin (*target*) mitkä ominaisuudet muuttuvat, kun tilaan saavutaan. *Shown*-tilassa muutetaan *hidabletools*-elementin läpinäkyvyys ja koko ykköskiksi, jolloin se näkyy täysikokoisena. *Hidden*-tilassa ne taas vaihdetaan nolliksi, jolloin elementti muuttuu näkymättömäksi ja hyvin pieneksi.

Tilamuutokset voidaan animoida näyttävästi määrittelemällä transitoita (transitions). Esimerkin `NumberAnimation` määrittelee *scale*- ja *opacity*-ominaisuudet muuttumaan vähitellen käyrän nimeltä `InOutBack` mukaisesti. Näin nappulat zoomaavat ilmestyessään ja kadotessaan. QML:n animaatio-ominaisuudet ovat laajat, joten eri transiiovaihtoehtoja kannattaa kokeilla dokumentaation avustuksella.

Nappi `hidebutton` näyttää ja piilottaa muut napit. C-kielestä tuttu `a?:b:c` -operaattori luetaan ”jos `a` on tosi, niin `b`, muuten `c`”. Sillä saa kätevästi ehtolauseen yhdelle riville.

Muut napit ovat `Row`-elementin nimeltä `hidabletools` sisällä. Ne asettavat `webview`:in `url`-ominaisuudeksi halutun sivun osoitteen.

```
import QtQuick 2.0
import QtWebKit 3.0

Item {
    width: 800
    height: 600

    WebTools {
        z: 10
        anchors.fill: parent
    }

    WebView {
        id: webview
        url: "http://skrolli.fi"
        anchors.fill: parent
    }
}
```

Listaus 4

```
import QtQuick 2.0

Item {
    id: webtools
    state: "SHOWN"
    states: [
        State {
            name: "SHOWN"
            PropertyChanges {
                target: hidabletools
                opacity: 1
                scale: 1
            }
        },
        State {
            name: "HIDDEN"
            PropertyChanges {
                target: hidabletools
                opacity: 0
                scale: 0
            }
        }
    ]
    transitions: Transition {
        NumberAnimation {
            properties: "scale, opacity"; easing.type: Easing.InOutBack
        }
    }
}

MyButton {
    id: hidebutton
    text: webtools.state == "SHOWN" ? "Hide" : "Show"
    z: 10
    onClicked: webtools.state = webtools.state=="SHOWN" ? "HIDDEN" : SHOWN
}

Row {
    id: hidabletools
    spacing: 20
    anchors.horizontalCenter: parent.horizontalCenter

    MyButton {
        text: "Skrolli.fi"
        onClicked: webview.url = "http://skrolli.fi"
    }
    MyButton {
        text: "Skrolli FB"
        onClicked: webview.url = "https://www.facebook.com/Skrollilehti"
    }
    MyButton {
        text: "Skrolli Twitter"
        onClicked: webview.url = "https://twitter.com/skrollilehti"
    }
}
}
```

Listaus 5

Qt Quick ja Qt Declarative

Nämä termit tulevat usein vastaan QML:n yhteydessä. Qt Quick tarkoittaa koko sovelluskehystä, jonka osana QML-kieli ja siihen liittyvät C++-luokat ovat. Qt Declarative taas on Qt Quickin osa, joka lataa QML-tiedoston ja piirtää sen ruudulle.

Qt Quick 1 ja 2

Qt Quick 2:n sisältävä Qt 5 julkaistiin joulukuussa 2012. Tämä artikkeli kirjoitettiin alun perin Qt 4.8:aa ja Qt Quick 1:tä varten, joten esimerkit toimivat myös Qt Quick 1:ssä hyvin pienin muutoksin. Qt Quick 2:n uusia ominaisuuksia ovat mm. OpenGL-varjostimet, joilla saadaan aikaiseksi silmäkarkkia, Canvas-elementti, johon voidaan piirtää 2D-grafiikkaa ja ikkunoiden hallinta. OpenGL-tukensa ansiosta Qt Quick 2 myös piirtää grafiikkaa nopeammin.

QtQuick3D

QtQuick3D:llä kaksiulotteisten QML-käyttöliittymien sekaan saadaan myös 3D-sisältöä, jota voidaan animoida yhtä helposti kuin 2D-sisältöäkin. Myös OpenGL-varjostimia tuetaan. QtQuick3D on näillä näkymin tulossa osaksi Qt 5:tä.



Esimerkki 3: Yksinkertainen peli

Lopuksi teemme klassisen Pong-pelin QML:llä. Tässä yksin pelattavassa versiossa mailaa ohjataan hiirellä ja jokaisesta torjutusta pallosta saa yhden pisteen. Luo Pong.qml ja kirjoita siihen listauksen 6 lähdekoodi.

Pongin juurielementtinä on musta nelikulmio. Paddle on pelaajan ohjaama maila, joka saadaan liikkumaan asettamalla sen y-ominaisuudeksi MouseArealta saatava hiiren y-koordinaatti. Jotta koordinaatti päivittyisi muulloinkin kuin nappia painaessa on hoverEnabled-ominaisuus kytkettävä päälle. Ball on pallo, jonka ominaisuudet dx ja dy merkitsevät pallon nopeutta X- ja Y-akselien suhteen.

Mielenkiintoisin elementti tässä esimerkissä on Timer, jolla suoritetaan JavaScript-koodia ajastettuna. Interval-ominaisuus on liipaisun väliaika millisekunteina, ja Repeat saa liipaisun toistumaan säännöllisesti. Liipaisussa toteutettava koodi on onTriggered-signaalinkäsittelijässä ja sisältää Pongin pelilogiikan JavaScript-koodina. Kaikki QML:n elementtien id:t, muutujat ja signaalit näkyvät JavaScript-koodiin. Pongia pääset pelaamaan lisäämällä sen esimerkiksi skrolli.qml:ään halutun kokoisena.

```
import QtQuick 2.0

Rectangle {
    color: "black"
    MouseArea {
        id: mouseArea
        hoverEnabled: true
        anchors.fill: parent
    }
    Rectangle {
        id: paddle
        width: parent.width/100
        height: parent.height/8
        x: width
        y: mouseArea.mouseY
    }
    Rectangle {
        id: topWall
        width: parent.width
        height: parent.height/40
    }
    Rectangle {
        id: bottomWall
        width: parent.width
        height: parent.height/40
        anchors.bottom: parent.bottom
    }
    Rectangle {
        id: backWall
        width: parent.width/40
        height: parent.height
        anchors.right: parent.right
    }
    Rectangle {
        id: ball
        width: parent.width/50
        height: width
        x: parent.width/2
        y: parent.height/2
        z: 10
        property real dx: 5
        property real dy: Math.random()*6-3
    }
    Text {
        id: scoreDisplay
        anchors.horizontalCenter: parent.horizontalCenter
        anchors.top: topWall.bottom
        font.pointSize: parent.height/20
        color: "white"
        property int score: 0
        text: score
    }
    Image {
        anchors.centerIn: parent
        source: "http://www.skrolli.fi/logo1.png"
        opacity: 0.5
    }
    Timer {
        interval: 16
        repeat: true
        running: true
        onTriggered: {
            ball.x += ball.dx
            ball.y += ball.dy
            // Törmäykset ylä- ja alaseiniin
            if(ball.y >= bottomWall.y-bottomWall.height ||
            ball.y <= topWall.y + topWall.height) ball.dy = -ball.dy
            // Törmäys takaseinään
            if(ball.x >= backWall.x - backWall.width) ball.dx = -ball.dx
            // Törmäys mailaan, pisteiden ja nopeuden lisäys
            if(ball.x <= paddle.x + paddle.width &&
            ball.y > paddle.y &&
            ball.y < paddle.y+paddle.height) {
                ball.dx = Math.abs(ball.dx)*1.1
                ball.dy *= 1.1
                scoreDisplay.score++
            } else if(ball.x < 0) { // Peliin häviäminen
                ball.dx = 5
                ball.dy = Math.random()*6-3
                ball.x = parent.width/2
                ball.y = parent.height/2
                scoreDisplay.score = 0
            }
        }
    }
}
```

Listaus 6

Linkkejä

Digia Qt SDK

<http://qt-project.org/downloads>

Qt Quick-dokumentaatio

<http://qt-project.org/doc/qt-5.0/qtquick/qtquick-index.html>

Tämän artikkelin lähdekoodit

<http://www.skrolli.fi/skrolli-qml.tar.gz>



Viestejä ylihuomisen lukijalle

Tulevaisuudessa avaruuskissat
ampuvat lasereita nänneistään, mutta
miltäpä maistui 2010-luvun alku?

Tapio Berschewsky

Scrollin julistettu tarkoitus on olla aikaa kestävä tekniikkamedia. Jokainen Scrollin numero on siis aikakapseli. Eräänlainen viesti heille, jotka poimivat toivottavasti jo aikanaan puhkiluetun lehden käteensä vuonna 2030, kun kellarin sisällöstä viimein ihan oikeasti pitää heittää turha pois. Heille haluan jättää ajankuvan — kertomuksen siitä millainen Suomi, maailma, ja internet oli, kun Scrollia ruvettiin tekemään.

Tämä teksti kirjoitettiin ankean pimeänä aamupäivänä tammikuussa 2013, myrskyisenä ajanjaksona internetin historiassa. Täysi taistelu verkon avoimuutta vastaan oli roihunnut maassamme jo useamman vuoden ajan. Sensuurin ja korporaatiodien oikeuksien puolesta toittottivat sokerisilla viesteillään poliittisella kentällä kaikki valtapuolueet.

Poliittinen tunnelmamme oli virittynyt polkemaan yksilön oikeuksia ja erityisesti viemään viimeisetkin mahdollisuudet yksityisyyteen arkipäiväisessä elämässä. Poliitikot, yritykset ja yhteisöt vaativat jatkuvasti suurempia oikeuksia ihmisten verkkoliikenteen julkeaan ja luvattomaan tutkimiseen.

Vielä vuonna 2012 oli Suomessa öykkärimäiselle yhdistykselle mahdollista käskyttää verorahoista maksetut auktoriteetit varastamaan pikkutyön tietokoneen ja kiristämään tämän isältä rahaa. Lelu vietiin kädestä, koska yhdistys oli vakoilemalla tytön yksityistä verkkoliikennettä pääteltyä tämän ladanneen tusinapoppitähden lauleskelua. Tätä on toivottavasti vielä vaikeampi ymmärtää siellä siintävässä tulevaisuudessa.

Seinän takana nussitaan

Yksityisten ihmisten mielivaltainen verkkosensuuri täytti Suomessa viisi vuotta näillä tienoin. Ajan hengen mukaisesti sitä kutsuttiin lapsipornosuodatuksi. Sensuurilista kuitenkin sisälsi myös sellaisia henkilökohtaisia sivustoja, joiden sisältö oli puhdasta poliittista kritiikkiä ilman pornon hiventäkään.

Matti Nikki perusti verkkosensuurin alkutaipaleilla lapsiporno.info -sivuston kertomaan listasta ja sitä koskevasta laisäädännöstä. Sivusto päättyi listalle välittömästi, vaikka se ei sisältänyt lainkaan lapsipornoa. Kryptofasistinen valtiovalta pyrki Suomessa estämään yksityisten ihmisten sananvapautta ja oikeutta omaan mielipiteeseen.

Koko sensuurilista kertoi paljon siitä kaksinaamaisuudesta

millä suomalaiset päättäjät toimivat. Varsinaiseen hyväksikäyttöön ei puututtu, vaan kukkahattutädin sekä ruman tapahtuman välille pystytettiin seinä, jossa oli poliisin logo. Lapsia raiskattiin yhä, mutta sitä ei enää tarvinnut ajatella, eikä poliitikkojen tarvinnut pystyä tekemään mitään itse ongelmalle.

Horisontissa paloi jo

Suomi oli nettifasisminsa kanssa hyvin linjassa muun maailman kanssa. Myös globaalisti pidettiin poppitähtösten, julkaisuyritysten ja etujärjestöjen miljardeja euroja tärkeämpänä kuin tavallisia yksilöitä ja heidän oikeuksiaan.

Noin vuosi ennen tämän kolumnin kirjoittamista kansainvälinen viranomaisten koalitio hyökkäsi Kim Dotcomin johtaman Megauploadin ja miehen itsensä kimppuun. Maailman ensimmäisiin kuuluva pilvipalvelu kaadettiin, koska se sisälsi myös tekijänoikeuksilla suojattua materiaalia.

Yhdysvaltalaisen mediamahdun käskystä Yhdysvaltain oikeusministeriö ja tiedustelupalvelu tekivät usean suvereenin valtion kanssa yhteistyötä vakoillakseen lainvastaisesti toisen maan yksityisen asukkaan toimintaa. Nämä tahot yhdessä hyökkäsivät hillittömällä aseellisella voimalla tämän asukkaan kotiin, riistivät häneltä hänen omaisuutensa, ja takavarikoivat kaiken datan palvelusta.

Megauploadin viimeisenä toimintavuonna palvelu kattoi 4 % koko internetin liikenteestä. Peräti 25 % yritysten internet-liikenteestä kohdistui Megauploadiin, joka oli suosittu varmuuskopiointipalvelu premium-asiakkaittensa parissa. Kaikki tämä miljoonien yksilöiden, tuhansien yritysten ja satojen valtiollisten elimien tieto varastettiin Yhdysvaltain oikeusministeriön omaisuudeksi.

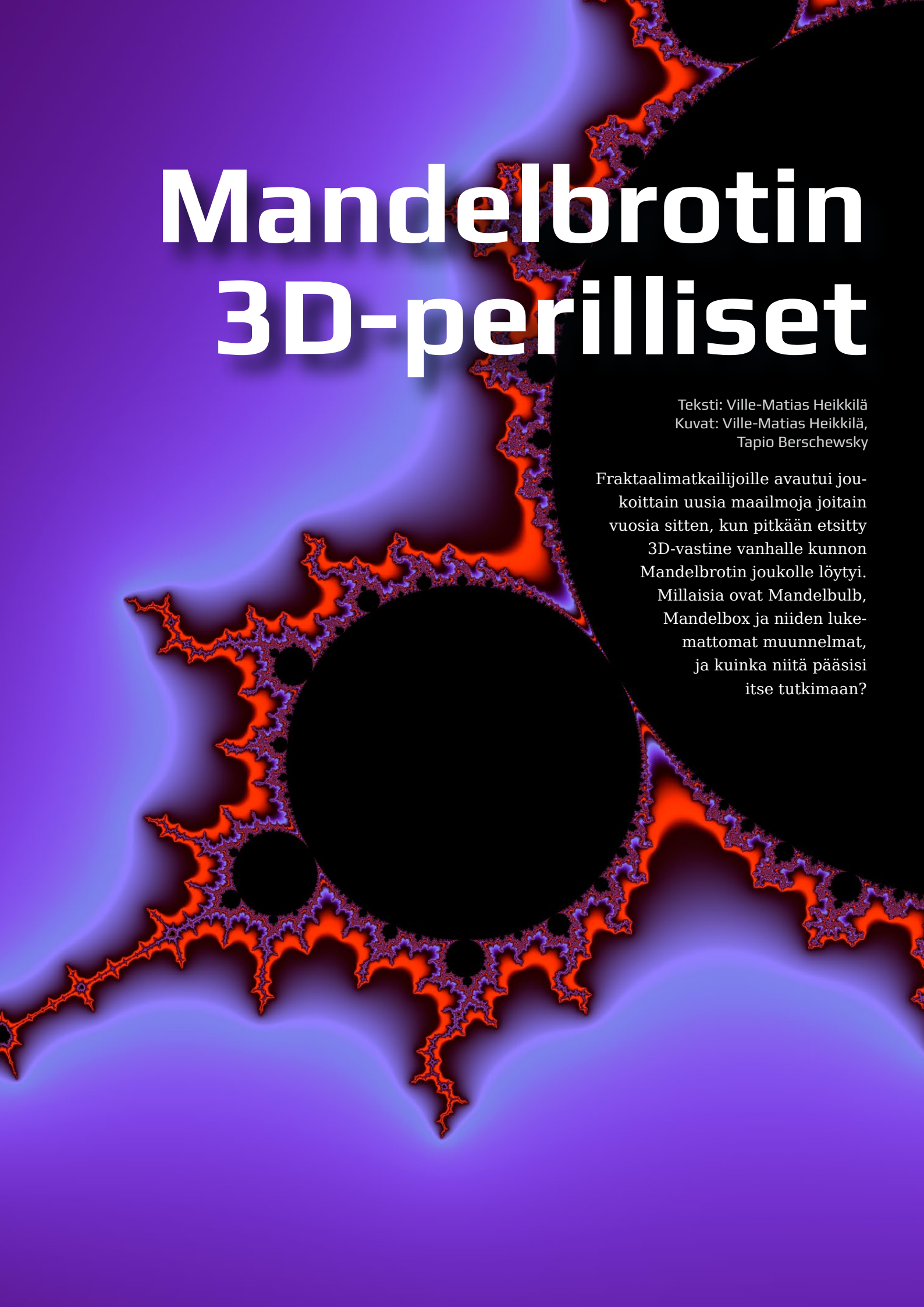
Tämä kaikki oli tietysti vain alkua. Siellä vuodessa 2030 olette jo päässeet kokemaan sen, kun maailmaan tulostettiin ensimmäiset aseet, räjähteet ja huumeet. Olette kokeneet sen paskamyrsdyn, joka syntyi, kun Monsanto patenttoituja geenejä printattiin synteettiseksi siemeneksi. Tiedätte, että tekijänoikeuksien vahtikoirat ja sensuurihimoiset uraohjuspoliitikot olivat vain ensimmäinen askel vanhan vallan ja korporaatiodien johtamassa sodassa tietokoneita ja erityisesti tietoverkkojen mahdollisuuksia vastaan.

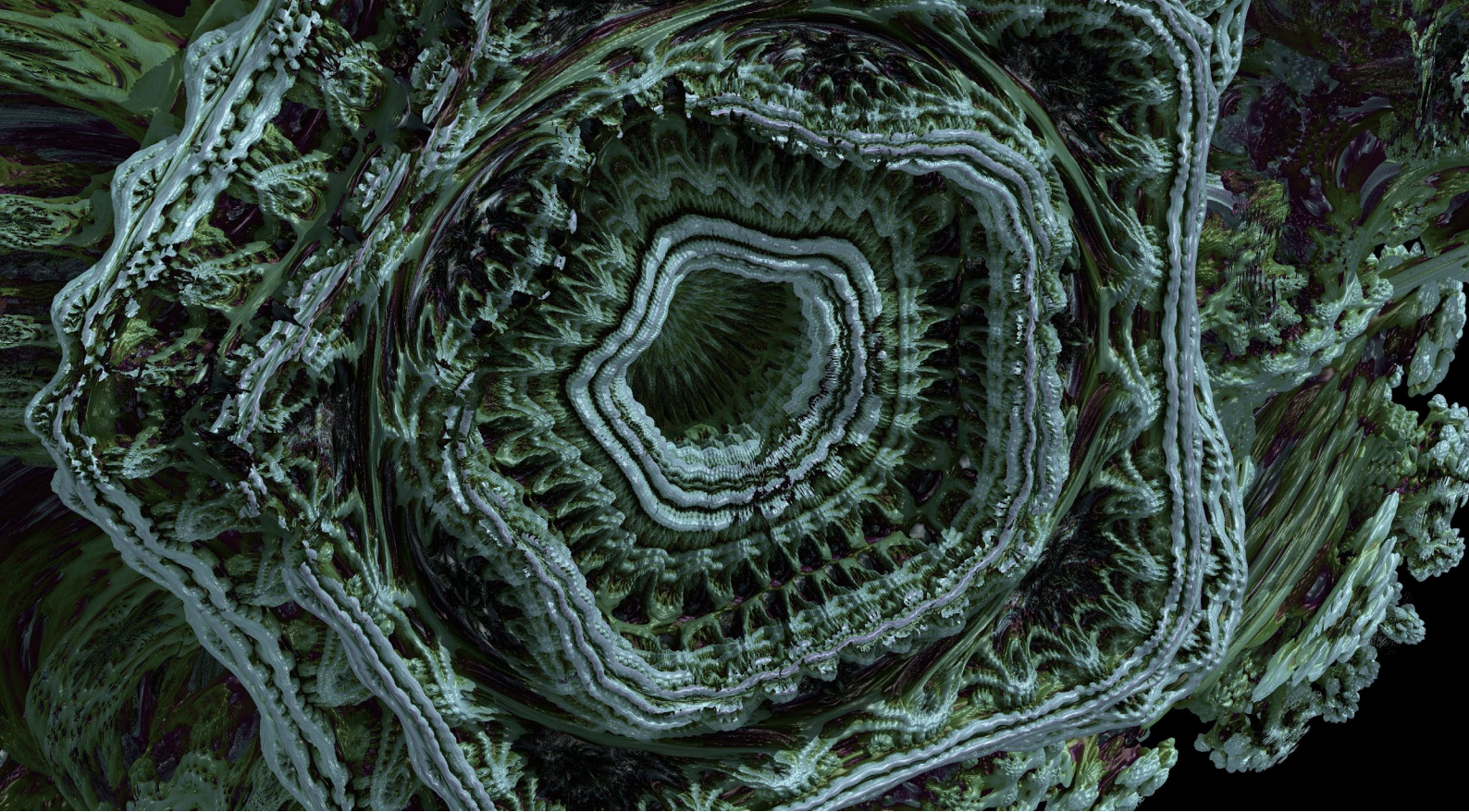
Meille se ymmärrys oli vasta leviämässä.

Mandelbrotin 3D-perilliset

Teksti: Ville-Matias Heikkilä
Kuvat: Ville-Matias Heikkilä,
Tapio Berschewsky

Fraktaalimatkailijoille avautui joukoittain uusia maailmoja joitain vuosia sitten, kun pitkään etsitty 3D-vastine vanhalle kunnan Mandelbrotin joukolle löytyi. Millaisia ovat Mandelbulb, Mandelbox ja niiden lukemattomat muunnelmät, ja kuinka niitä pääsisi itse tutkimaan?





Useimmat lukijat varmasti tunnistavat näöltä Mandelbrotin joukon — kaksiulotteisen kuvion, jossa hieman lumiukkoa muistuttavasta keskuspalllostosta erkanee joka suuntaan loputtomasti haarautuvia salamaisiakin ulokkeita. Se lienee kaikkien aikojen suosituin fraktaaliksi, joka on ehtinyt kiehtoa ihmismieliä jo kolme vuosikymmentä.

Mandelbrotin joukon löytäjinä voidaan pitää ranskalaisia Gaston Juliaa ja Pierre Fatouta, jotka tutkivat sen perustana olevaa matematiikkaa liki sata vuotta sitten. Kuvion muoto on kuitenkin turhan työläs laskea käsin. Siitä saatiinkin aavistus vasta kun amerikanpuolalainen Benoît Mandelbrot ja eräät muut matemaatikot ottivat avuksi tietokoneet 1970-luvun lopulla. Muutamien vuosien kuluttua Mandelbrotin ja Julian joukkojen kuvia päätyi myös kansantajuisiin tiedelehtiin ja sitä kautta tietokoneharrastajien ulottuville.

1980-luvun lopulla kotitietokoneille alkoi ilmestyä ohjelmia, joilla kuka tahansa pääsi kaivautumaan Mandelbrotin joukon uumeniin. Kuvien valmistumistakaan ei tarvinnut enää odottaa tunteja. Myös fraktaalikaavoja tuli ohjelmiin lisää pioneerihenkisimpien harrastajien kokeilujen tuloksena: suosituksessa Fractint-ilmaisohjelmassa oli 1990-luvun puolivälissä jo satakunta kaavaa lukemattomine säädettävine parametreineen. Fraktaali-innostus oli kuitenkin tuolloin jo hiipumassa — fraktaalilithan olivat vain tylsää 2D-grafiikkaa, ja päivän sana oli kolmiulotteisuus!

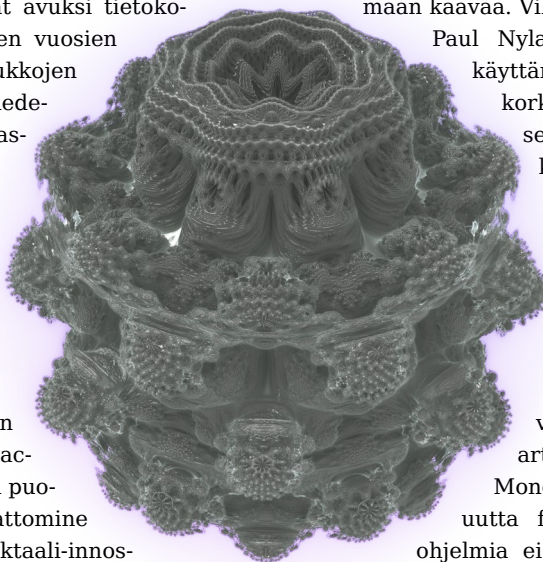
Toki kolmiulotteista fraktaaligrafiikkaa on ollut olemassa jo pidemmän aikaa. 2D-fraktaalit saattoi helposti piirtää korkeuskarttavuoristoiksi, ja Sierpinskiin kolmiot muuntuivat hyvinkin kivuttomasti Sierpinskiin pyramideiksi. Mandel-

brotin joukon kaltaisille kompleksilukufraktaaleille taas ei tuntunut löytyvän tyydyttävän näköisiä 3D-vastineita milään. Vaikka esimerkiksi 4D-juloiden 3D-leikkausten muodonmuutokset olivatkin mielenkiintoisia, eivät ne vedonneet kauneudentajuun alkuperäisten 2D-juloiden pyörteiden tavoin.

Marraskuussa 2007 päästiin vihdoinkin aidon 3D-Mandelbrotin jäljille, kun fraktaaliharrastaja Daniel White ehdotti fractalforums.com-sivustolla uutta tapaa laajentaa Mandelbrotin kompleksilukumatematiikka kolmiulotteiseksi. Ensimmäiset näin aikaansaadut kuvat olivat edelleen pienoinen pettymys, mutta idea innosti silti monet muuntelemaan kaavaa. Vihdoin elokuussa 2009 matemaatikko

Paul Nylander löysi jotain hätkähdyttävää: käyttämällä neliöön korotuksen sijaan korkeampia potensseja alkoi kappaleeseen ilmestyä kasvimaisia ulokkeita, kiemurtelevia seittejä ja groteskeja suuaukkoja. Kahdeksannen potenssin kappale, jota voisi luonnehtia vaikka parsakaalin ja Giger-henkisen avaruusaluksen sekasikiöksi, sai nimekseen Mandelbulb.

Todellinen kultaryntäys alkoi, kun Daniel White julkaisi kotisivullaan Mandelbulbia käsittelevän artikkelin mieltä kiehtovine kuvineen. Monet olivat innokkaita tutkimaan tätä uutta fraktaalikappaletta, mutta valmiita ohjelmia ei ollut. Siispä aiheesta innostuneet päätyivät kirjoittamaan omia renderöijäänsä ja myös kokeilemaan niissä erilaisia vaihtoehtoisia kaavoja ja muunnelmia. Puolalainen Krzysztof Marczak kehitti omasta renderöijästään monipuolisen työkalun, jonka hän julkaisi nimellä Mandelbulber, ja jolla tämänkin jutun 3D-fraktaalikuvat on tuotettu.





Kuinka se toimii?

Vaikka fraktaaleja voikin tutkia valmis-ohjelmilla niiden matematiikkaa ymmärtämättä, on tutkimusmatkailu antoisampaa, jos kaavojen toimintaperiaatteen jotenkin hahmottaa. Varsinkin Mandelbulber-ohjelma on täynnä parametreja, joita voi räplätä summanmutikassakin, mutta arvailut johtavat parempiin tuloksiin, jos käyttäjällä on jokin käsitys siitä, mitä nappuloista tapahtuu.

Mandelbrot-tyyppisissä fraktaaleissa on pohjimmiltaan kyse siitä, mitä luvuille — esimerkiksi pisteen x - ja y -koordinaateille — tapahtuu, kun niihin sovelletaan toistuvasti samaa laskukaavaa. Monet kaavat ovat suhteellisen tylsiä, esimerkiksi neliöön korotus: luvusta tietää jo etukäteen, karkaako se itsellään toistuvasti kerrottaessa äärettömyksiin. Benoît Mandelbrotin tutkima kaava on sen sijaan ennustamattomampi: siinä karkaamisraja muodostaa loputtomasti kiemurtelevan fraktaalimuodon, jonka löytämiseksi on kaavaa toistettava paikoin jopa kymmeniä kertoja.

Mandelbrotin toistokaava on lyhykäisyydessään $z \leftarrow z^2 + c$. z ja c ovat yksiuotteisten reaali- ja kompleksilukujen sijaan kaksiuotteisia kompleksilukuja. Vakio c sisältää pisteen koordinaatit, ja z on muuttuja, jonka arvo on alussa sama kuin c . Perinteiset 2D-fraktaaliohjelmat käyvät läpi kuva-alueen jokaisen pikselin ja soveltavat niiden koordinaateille tätä kaavaa. Pikselin väri määräytyy yleensä pakojasta: siitä, kuinka monta kertaa kaavaa käytetään, ennen kuin koordinaatit karkaavat kuvion ulkopuolelle eli täyttävät

ns. bailout-ehdon. Jottei ohjelma juuttuisi ikuisen silmukkaan, on kierros- eli iteraatiomäärälle asetettava yläraja. Mitä korkeammaksi tämä raja asetetaan, sitä pienempiä yksityiskohtia saadaan esiin, ja sitä pitempään kuvan piirtäminen kestää.

Julian joukoiksi kutsutut muodot syntyvät, kun toistokaavaa käytetään tavalla, jossa c on koko kuvion alueella sama. Näin esimerkiksi jokaisesta Mandelbrotin joukon pisteestä käsin voidaan laskea oma Julian joukkonsa. Varsinkin vanhoissa ohjelmissa esiintyy fraktaalityyppi nimeltä ”Julia” viitaten nimenomaan Mandelbrotin joukon Julia-vastineisiin, mutta Julian joukkoja voidaan johtaa muillekin fraktaalille, joiden toistokaavaan kuuluu samantapainen vakionsummaus (”+ c -osa”).

Useimmat ohjelmointikielet eivät tue kompleksilukuja suoraan, ja ihmisillekin kaava on helpompi ymmärtää, jos se kirjoitetaan auki x - ja y -koordinaateilla tapahtuviksi laskutoimituksiksi. Kun kaavaan $z \leftarrow z^2 + c$ sijoitetaan z :n paikalle $x + yi$ ja c :n paikalle $x_0 + y_0i$, missä i on kompleksilukujen määritelmän mukainen -1 :n neliöjuuri, saadaan välivaiheiden jälkeen:

$$\begin{aligned}x &\leftarrow x^2 - y^2 + x_0 \\y &\leftarrow 2xy + y_0\end{aligned}$$

Yksi syy sille, että Mandelbrotin joukon 3D-vastine oli niin vaikea löytää, oli ettei kompleksiluvuille ole aitoa kolmiuotteista vastinetta. Kompleksiluvut piti siis korvata jollain, joka tekisi samantapaiset

laskutoimitukset kolmella koordinaatilla. Useimmat lähestyivät tätä ongelmaa liian aritmeettisesti, kunnes Daniel White keksi ajatella asiaa geometrisemmalta kannalta. Kun kompleksiluku kerrotaan itsellään, se nimittäin kiertää nollapistettä niin, että luvun ja nollan välinen kulma kaksinkertaistuu. Samalla myös luvun etäisyys nolasta korottuu toiseen potenssiin. Whiten käyttämässä napakoordinaattiratkaisussa muotoa $z \leftarrow z^n + c$ olevat toistokaavat kääntyvät muotoon

$$\begin{aligned}x &\leftarrow r^n \sin(na) \cos(nb) + x_0 \\y &\leftarrow r^n \sin(na) \sin(nb) + y_0 \\z &\leftarrow r^n \cos(na) + z_0\end{aligned}$$

missä (r, a, b) on edellisen kierroksen (x, y, z) :n napakoordinaattiesitys.

Kuvat Mandelbrot-tyyppisistä 3D-fraktaaleista tuotetaan yleensä raymarching-menetelmällä, joka on erikoistapaus raytracingista eli säteen-seurannasta. Menetelmässä kuva-alan kullekin pisteelle lasketaan kuvitteellinen valonsäteen saapumislinja ja edetään sitä pitkin askel kerrallaan pois päin, kunnes toistokaavan käyttö osoittaa, että on törmätty fraktaalien seinämään. Tarvitavien askelten pituuksien arvioimiseksi johdetaan toistokaavasta yleensä sopiva etäisyydenarvioimiskaava. Törmäyskohdassa lasketaan vielä usein pinnan normaalivektori varjostuksia ja heijastuksia varten. Ohjelmoinnillisista yksityiskohdista ja optimointitempuista kiinnostuneiden kannattaa tutustua 3D-fraktaalien kanssa paljon puuhailleen demo-ohjelmoi- ja Iñigo Quilezin kotisivuihin.

Mandelbox ja muut muunnelmät

Mandelbulb on melko raskas fraktaalilaskettavaksi runsaan koordinaattien edestakaisin muuntelun vuoksi. Mandelbulbin löydyttyä monet grafiikkaohjelmoijat ryhtyivätkin etsimään kevyempiä kaavoja, jotka tuottaisivat yhtä mielenkiintoista jälkeä. Menestyksekkäs löytö oli Tom "Tglad" Lowen Mandelbox, jolla saa aikaan teknisiä, futuristisia ja rakennustaiteellisia näkymiä.

Mandelboxin toistokaavassa tärkeä käsite on koordinaattien taittaminen eli folding, jossa koordinaatit ikään kuin pyritään pitämään origokeskisen kappaleen sisällä mutta sen verran huonosti, että riittävän kauas joutunut piste karkaa kaikesta huolimatta ulos. Toistokaavaan kuuluu kolme vaihetta: kuutiotaittovaiheessa pistettä sovitetaan kuution ja pallotaittovaiheessa pallon sisään, ja skaalausvaiheessa sen koordinaatit kerrotaan vakioilla ja niihin lisätään alkupeiraisen testattavan pisteen koordinaatit. Mitään standardimuotoa Mandelboxilla ei ole, mutta erilaisilla vakioilla kappaleen saa näyttämään niin borg-kuutiolta, futuristiselta kaupungilta kuin epämääräiseltä silpultakin.

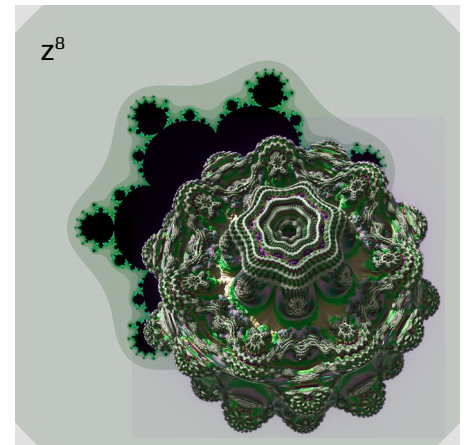
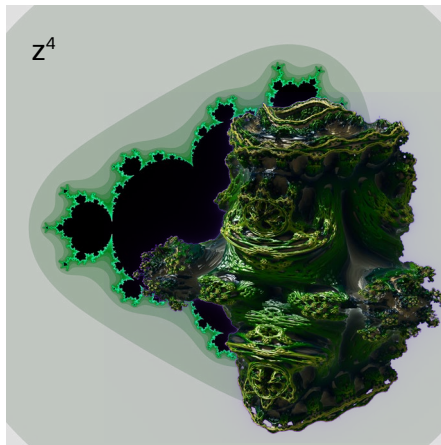
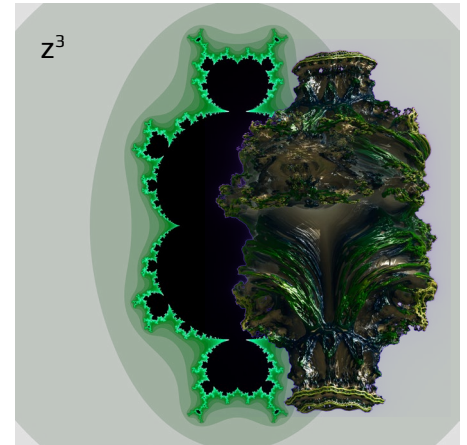
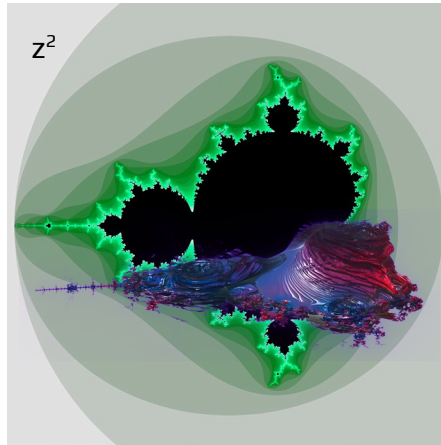
Pian Mandelboxin löydyttyä Mandelbulberin kehittäjä Krzysztof Marczak huomasi, että Mandelbox-kaavan taittovaiheiden yhdistäminen Mandelbulb-kaavaan tuottaa mielenkiintoisia yhdistelmäfraktaaleja. Muodoista, joissa Mandelboxin pallotaittovaihe on korvattu Mandelbulbin toistokaavalla, on joskus käytetty nimitystä BulbBox.

Marczak on kehittänyt Mandelboxista myös pyöreämpimuotoisen muunnelman, joka löytyy Mandelbulberista nimellä Smooth Mandelbox. Siinä Mandelboxin joko-tai -tyyppiset vertailut on korvattu pehmeämmillä siirtymillä, jotka tuovat kaavaan muutaman säädettävän parametrin lisää.

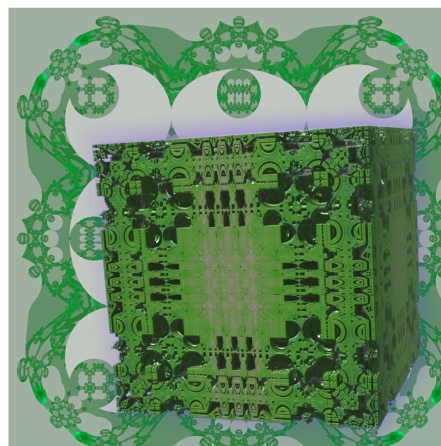
Mandelbrotin lisäksi myös muita kaksikulotteisia pakoikafraktaaleja on kokeiltu kolmiulotteistaa Whiten kaavalla. Kaikkiin se ei toimi kovin hyvin, mutta esimerkiksi vuonna 1992 löydetty Burning Ship muuntuu sillä mielenkiintoisen näköiseksi temppelikaupungiksi.

Mandelbrotin perillisiä on myös risiinsiitetty aivan toisenlaisten matemaattisten muotojen, IFS-fraktaalien kanssa. Esimerkiksi Sierpinskiin pyramidista saa vähemmän tylsän, kun sen yhdistää Mandelboxin pallotaittoon.

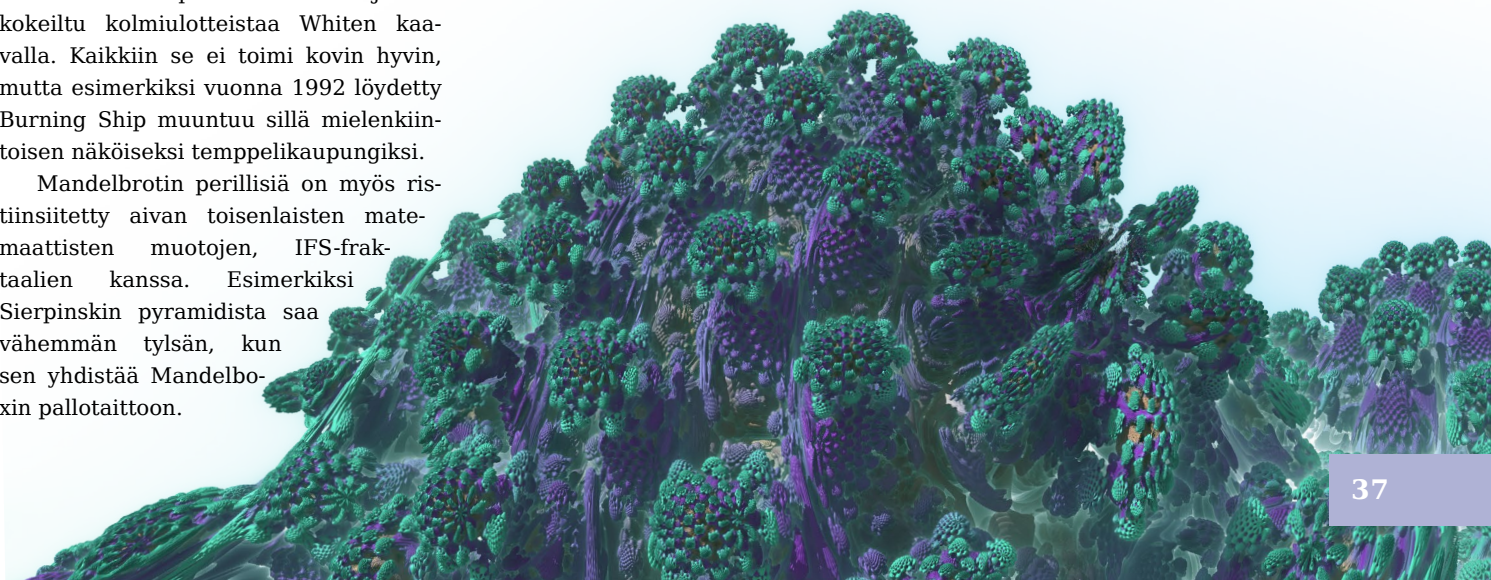
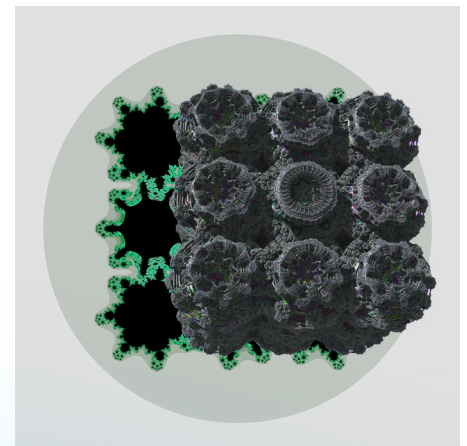
Mandelbrot / Mandelbulb



Mandelbox



Bulbbox



Mandelbulber-ohjelman käyttö

Mandelbulber-ohjelman saa sen kotisivulta osoitteesta mandelbulber.com. Se on tarjolla paitsi lähdekoodimuodossa, myös valmiiksi käännettyinä versioina Windowsille, Linuxille ja Macille. Tässä jutussa käsiteltävä versio 1.12.1 suorittaa laskentansa pääprosessoreilla, joten käytettävän näytönohjaimen suorituskyky ei nopeuta sitä. Vaihtoehtoinen ohjelmisto on Mandelbulb3d, joka on saatavilla vain Windowsille.

Mandelbulberin käyttöliittymä on ohjelman kokeellisesta luonteesta johtuen melko sekava, ja koska ohjelma on suunnattu uteliaille fraktaaliharrastajille, ei tarve parantaa sen ymmärrettävyyttä liene ollut niin suuri kuin tarve lisätä uusia kikkoja ja yhdistelymahdollisuuksia. Sen pintapuolinen kokeilu on kuitenkin aika helppoa.

Ohjelmaan latautuvat oletusarvoisesti käynnistettäessä parametrit, joilla saa renderöityä Mandelbulbista kokonäkymän. Kuva piirtyy renderöinti-ikkunaan, kun ohjelman käynnistämisen jälkeen napsauttaa käyttöliittymäikkunan ylänurkassa olevaa render-painiketta. Yksityiskohtia pääsee katsomaan lähempää klikkailemalla mielenkiintoisia kohtia renderöinti-ikkunasta. Takaisin kokokuvaan pääsee view-välilehden reset view -painikkeella.

Ohjelman mukana tulee lukuisia esimerkkejä, joiden parametrit voi ladata käyttöliittymäikkunan alareunan load example -painikkeella. Parametritiedoston lataamisen jälkeen on kuva vielä renderöitävä render-painikkeella. Esimerkkejä voi ja kannattaakin käyttää pohjana omille kokeiluilleen — esimerkiksi

mielekkäiden IFS-parametrien keksiminen on melkoista lottoa, jos sen takana oleva matematiikka ei ole hallussa.

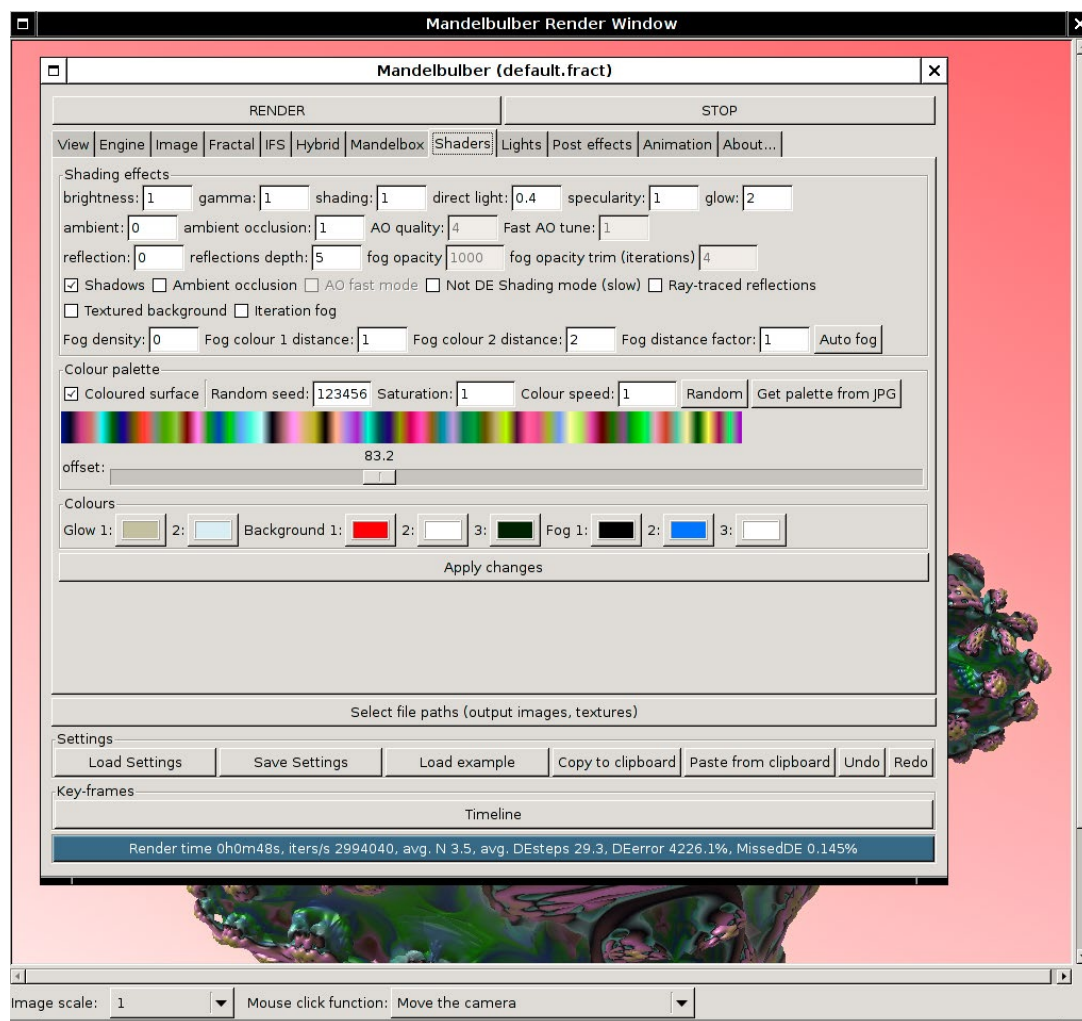
Fraktaalityyppi ja sen perusparametrit valitaan Fractal-välilehdeltä. Useilla tyypeillä on omia lisäparametrejaan, jotka löytyvät fraktaalien nimen mukaiselta välilehdeltä. Fractal-välilehden Folding-otsikon alta voi myös yhdistää Mandelbox-taittokaavat mihin tahansa fraktaaliin — esimerkiksi aiemmin mainittu BulbBox saadaan aikaan valitsemalla fraktaalityypiksi Mandelbulb ja kytkemällä kuutiotaitto eli "Tglad's folding mode" päälle. Jos kamerakulma jää fraktaalityypin vaihtamisen jälkeen huonoksi, sen saa usein palautettua tolkkulliseksi reset view -toiminnolla.

Etenkin erilaisia parametreja kokeiltaessa ja sopivia kamerakulmia etsittäessä on kuvan koko hyvä vaihtaa Image-välilehden alta pienemmäksi, jottei vastetta tarvitse odottaa liian pitkään. Renderöinti-ikkunan alalaidasta voi myös valita pienelle kuvalle isomman skaalauskerroimen. Esimerkiksi 200x200-pikselisestä kuvasta skaalauskerroimella 4 saa jo käsityksen siitä, onko näkymä kiinnostava ja kamerakulma kohdillaan.

Renderöinnin yhteydessä kuvasta tuotetaan myös pikseli-kohtaista varjostusdataa, minkä ansiosta kuvan värejä, tehosteita, heijastusten ja varjostusten voimakkuuksia ja muita parametreja voi säätää tarvitsematta tuottaa koko kuvaa uusiksi. Nämä säädöt löytyvät Shaders- ja Post effects -välilehdeltä. Monet esimerkeistä käyttävät syväterävyydesthostetta (depth of field), joten jos kuva näyttää sumealta, kannattaa käydä kyt-

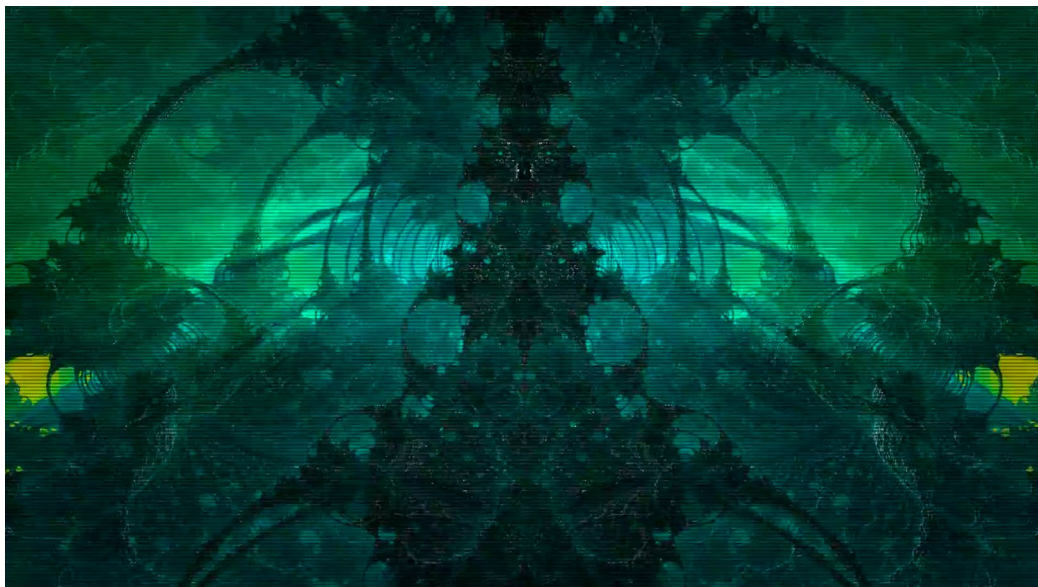
kemässä se pois päältä Post effects -välilehdeltä.

Tämä oli hyvin pikainen ja pintapuolinen katsaus Mandelbulberin käyttöön — esimerkiksi valaistus- ja animaatio-ominaisuudet jätettiin kokonaan käsittelemättä. 3D-fraktaalimaailmojen tutkiskeluun pääsee kyllä uppoutumaan varsinkin perusteellisesti ilmankin.



Onnistuu se reaaliajassakin!

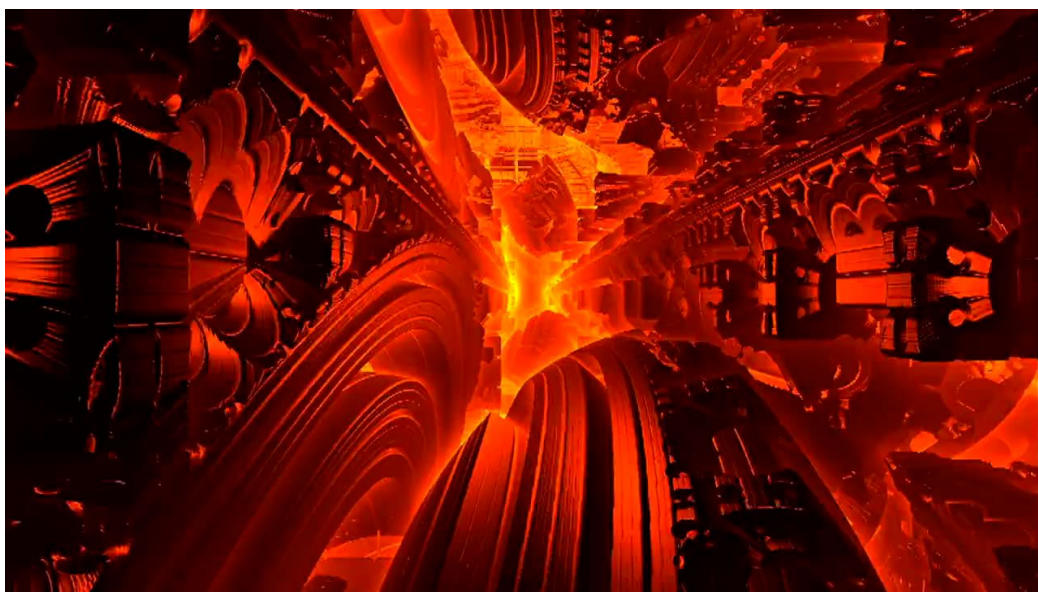
Samoin kuin reaaliaikaiset Mandelbrot-zoomaajat olivat valttia 1990-luvun puolenvälin demoskenessä, on nyt 2010-luvulla esiintynyt varsinkin pienikokoisissa PC-demoissa reaaliaikaista lentelyä laskennallisesti vaativissa 3D-fraktaaleissa. Nämä on käytännössä poikkeuksetta toteutettu shader-ohjelmointina, ja sulava näytönpäivitys vaatii näytönohjaimelta kovaa numeronmurskauskapasiteettia. Kohtuullisemman laitteiston käyttäjille on näistä demoista onneksi tarjolla myös videoversiot esimerkiksi YouTubessa.



Akronyme Analogiker:
HARTVERDRAHTET (4K, Win32)

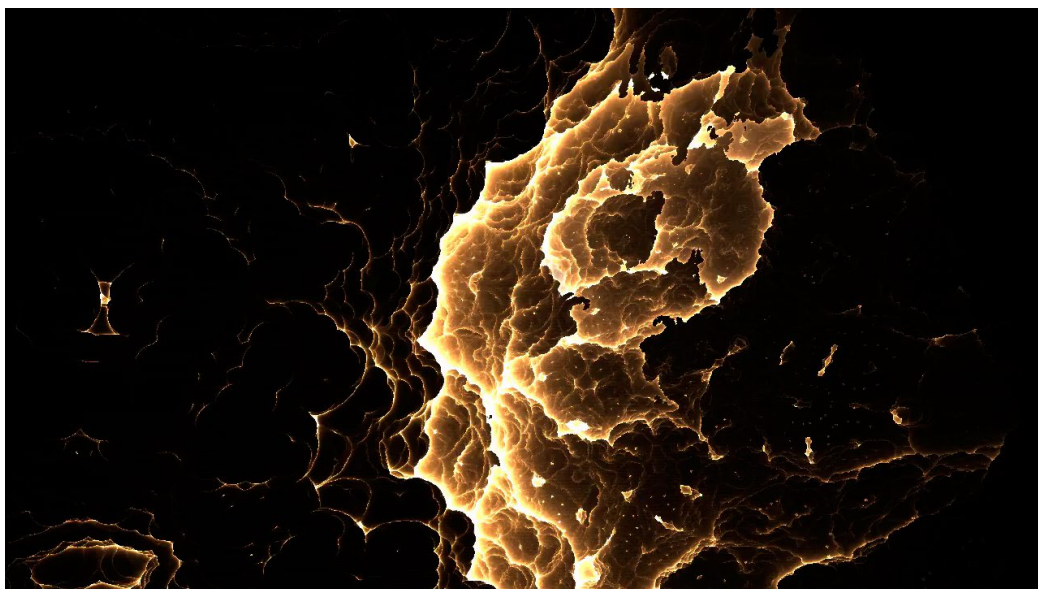
3D-fraktaalilentelyjä oli nähty jo ennen tätä Revision 2012 -tapahtuman 4K-demokilpailun voittanutta saksalaistuo-
tosta, mutta ne haeskelivat vielä tekniikkaa. Hartverdrahtet sen sijaan esittää fraktaalimaisemansa poikkeuksellisen tunnelmallisella ja mielikuvi-
tuksellisella tavalla.

Itse fraktaalikaava on Theli-At-tyyppinen hybridi, jonka parametrit muuttuvat kohtausten välillä.



TDA: EMBERS (1K, OSX)

The Digital Artistsin Embers osoittaa selkeästi, että yksi kilotavu riittää Mandelbox-lentelylle Mac OS:llä: fraktaalityyppi on helposti tunnistettavissa, eikä edes sen parametreja muunnella lentelyn aikana. Embers voitti Assembly 2012 -tapahtuman 1K-demokilpailun.



TBC: TRACELESS (1K, Win32)

1K-demon kakkossijalle pääsyt TBC:n Traceless käyttää kilotavun tilan edistyneemmin kuin voittaja ja on muutenkin parempi esitys. Demo sisältää monipuolisen oloisen fraktaaliraytracerin ja tunnelmallisen ääniraidan, ja sen lentoradat ja valokonstrastit miellyttävät silmää. Fraktaali muuntuu jatkuvasti ja siinä on havaittavissa eri vaiheissa sekä Sierpinski-tyyppisiä että orgaanisempia rakenteita.



Neuvostoliiton ja itäblokin tietokoneiden historiaa

ES EVM -sarjan tietokone ES-1035 käytössä Itä-Saksassa

Jotta edesmenneen itänaapurimme tietotekniikan historiaan johdatteleva lukuhetki olisi autenttisen nautinnollinen, suosittelemme Neuvostoliiton kansallishymnin soittamista taustalla, volyymi sopivasti kaakkoon kumartaen.

Jari Jaanto, Ville-Matias Heikkilä
Kuvat: Wikimedia Commons

Neuvostoliitto oli pioneeriassmassa useissa tieteen ja tekniikan alan harppauksissa. Se lähetti ensimmäisenä valtiona satelliitin maata kiertävälle radalle, ensimmäisen ihmisen avaruuteen, ja ensimmäisen pitkään asutetun avaruusaseman. Neuvostoliitto panosti tieteelliseen ja tekniseen tutkimustyöhön voimakkaasti, ja länsimaat saivat kilpailla tosissaan sitä vastaan.

Tietotekniikan kehitykseen Neuvostoliitto hyppäsi mukaan 40–50-lukujen taitteessa. Manner-Euroopan ensimmäinen ohjelmoitava elektroninen tietokone oli vuonna 1950 käynnistetty laskukone MESM (Малая Электронно-Счетная Машина), joka rakennettiin 1948–1950. Sotateollisuudella oli jo käytössään elekt-

ronisia laskinsovel-luksia ja analogisia tietokoneita, mutta digitaalisten tietokoneiden aikakausi alkoi MESM:stä. Se koostui 6 000 elektroniputkesta ja kulutti sähköä 24 kW. Sen laskuteho oli n. 50 laskutoimistusta sekunnissa ja sitä käytettiin huip-pusalaisissa ydin-aseprojekteissa.

Setun

Itätietotekniikan alkutaipaleella nähtiin monia innovaatioita. Yksi erikoisimmista koneista oli Moskovan yliopiston tutkimuskäyttöön 1958 valmistunut Setun (Серунь). Se käytti lukujärjestelmänään tavanomaisen binäärin sijasta ns. tasapainotettua trinäärijärjestelmää, johon kuuluu nollan ja positiivisen ykkösen lisäksi negatiivinen ykkönen.

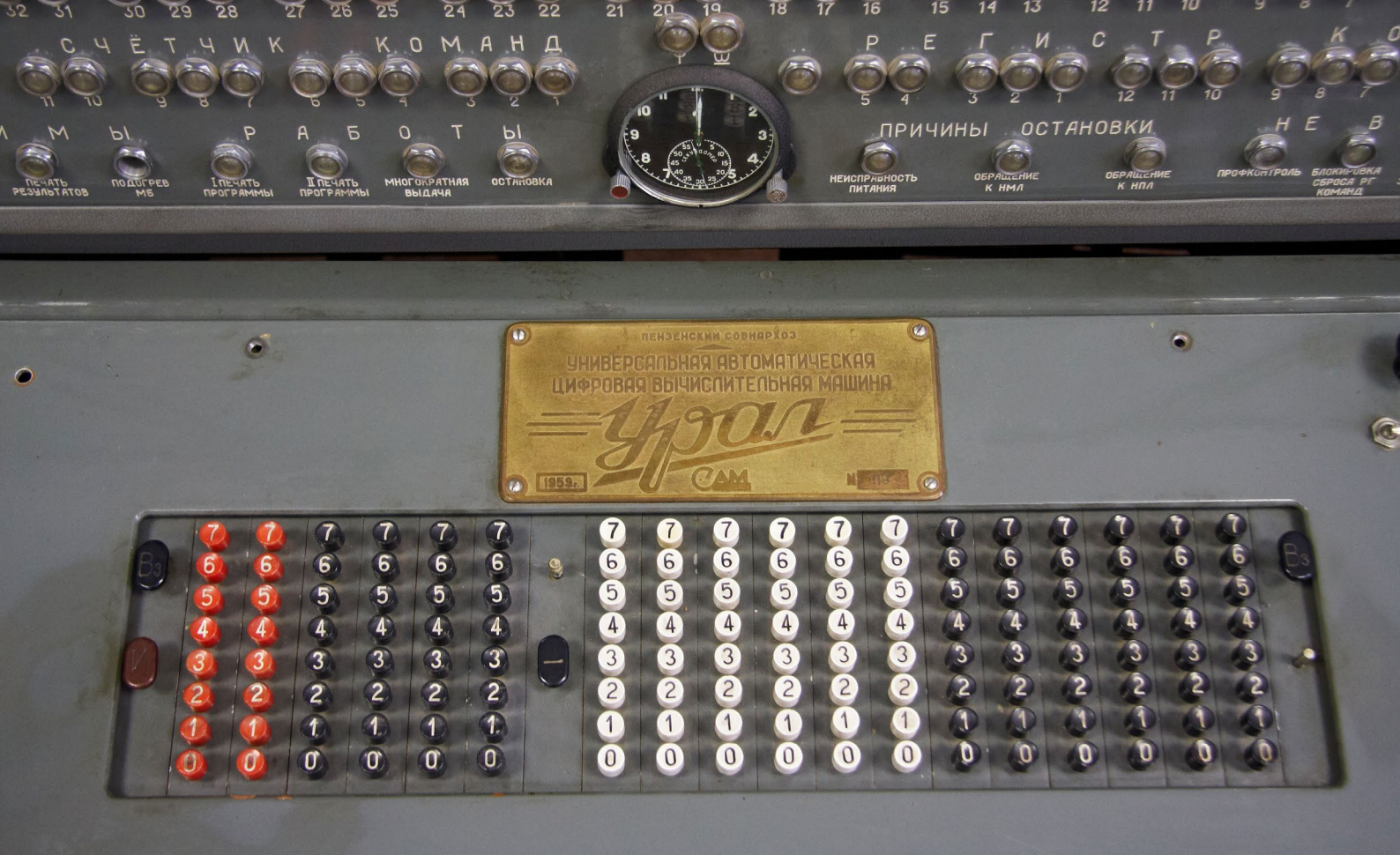
Setun oli teknisesti hyvin onnistunut. Tätä pidettiin etenkin trinäärijärjestelmän ansiona. Se osoittautui luotettavaksi ja vakaaksi erilaisissa lämpötiloissa ja volttimäärissä ja oli yksinkertainen rakentaa ja ajaa. Koneita valmistettiin n. 50 kappaletta vuoteen 1965 mennessä.



Setun

Luvut -6:sta +6:een desimaalina, binäärinä ja tasapainotettuna trinäärinä

-6	1010	-+0
-5	1011	-++
-4	1100	0--
-3	1101	0-0
-2	1110	0-+
-1	1111	00-
0	0000	000
+1	0001	00+
+2	0010	0+-
+3	0100	0++
+4	0101	+-+
+5	0110	++-
+6	0111	+++



Ural-1, neuvostotietotekniikan huippua 1950-luvulta

Vaikka Setun herätti kiinnostusta länsimaita myöten, pitivät suunnitelmatalouden päättäjät sitä kerettiläisenä, ja hanke jäädettiin. Moskovian yliopiston Setun korvattiin yhtä nopealla binäärikoneella, mutta sen kustannukset osoittautuivat moninkertaisiksi.

Setun sai vuonna 1970 seuraajan, Setun-70:n, jota ohjelmoitiin Forthin kaltaisella DSSP-kielellä. Setun ja Setun-70 jäivät maailman ainoiksi trinääritietokoneiksi.

Argon-16

1960-luvulla kehitys haarautui siviili- ja sotilaspolkuun, joista jälkimmäiseen avaruustietotekniikka kuului. Hävittäjät, avaruusalukset ja valvonta-asemat tarvitsivat kevyitä, vikasetoimia ohjaustietokoneita, joita edusti etenkin Argon-konesarja. Tunnetuin sarjasta on Argon-16 (Аргон-16). Se oli käytössä kaikissa Sojuz- ja Progress-avaruusaluksissa sekä Saljut-, Almaz- ja Mir-avaruusasemilla 37 vuoden ajan. Kone ei koskaan pettänyt, mikä tekee siitä luotettavimman koskaan avaruuskäytössä olleen tietokonemallin. Se on myös maailman pisimpään avaruuskäytössä palvellut tietokone.

Argon-16:n tuotanto aloitettiin 1974. Sen kaikki komponentit, muisti mukaan lukien, olivat kolmena samanaikaisesti toimivana kappaleena. Koneen tärkeimpiä suunnittelukriteereitä olivat luotettavuus ja reaaliaikainen yhteistyö avaruusaluksen muiden laitteiden kanssa.

Sojuz-aluksissa on vuodesta 2010 saakka käytetty Argonin sijaan uutta ohjaustietokonetta nimeltään TsVM-101 (ЦВМ-101).

Länsitietotekniikan kopiointi

Vuonna 1966 taloussuunnittelijat ehdottivat kehitettäväksi tietokoneiden sarjaa, johon kuuluisi laiteratkaisuiltaan vaihtelevia, keskenään ohjelmistoyhteensopivia konemalleja. Syntyi



Tämäkään Sojuz-lento ei olisi onnistunut ilman Argon-ohjaustietokonetta



Elektronika 60M



BK-0011M

elektronisten tietokoneiden yhtenäisjärjestelmä ES EVM (EC ЭВМ), jonka tuotanto alkoi 1972. Sarjan koneita valmistettiin niin Neuvostoliitossa kuin muissakin sosialistimaissa, ja se jatkui vielä Neuvostoliiton hajottua vuoteen 1998 asti.

Yhtenäisjärjestelmä mahdollisti ensimmäistä kertaa yhteensopivuuden länsimaisten koneiden kanssa. Vaikka järjestelmän pohjaksi oli tarjolla aivan päteviä itäkoneita, otettiin perustaksi amerikkalainen IBM S/360. IBM ei katsonut tätä pahalla vaan aloitti Neuvostoliiton kanssa yhteistyöneuvottelut, jotka katkesivat vuonna 1979 USA:n talouspakotteiden vuoksi.

Muita länsitekniikkaan perustuvia konesarjoja olivat SM EVM (PDP-11 ja VAX) ja 1980-luvun puolella ES PEVM (IBM PC). Keskeisten tuotantolinjojen ulkopuolella kloonattiin myös muita laitteita, kuten Apple II, Oric ja ZX Spectrum. Intelin 8080-suorittimen kloonaminen mahdollisti CP/M-yhteensopivat koneet. Myös supertietokone Cray-1 saatiin kloonattua 1980-luvun lopussa.

Mikrotietokoneet

Kun Intel toi 1971 markkinoille maailman ensimmäisen mikroprosessorin, ei Neuvostoliitossa oltu vielä paljoa jäljessä: K145IP1-suoritin kehitettiin vuonna 1973 ja ensimmäiset sitä käyttävät taskulaskimet tulivat myyntiin vuonna 1974. Vaikka neuvostojohto panosti taskulaskinten tuotantoon, se ei ollut kiinnostunut henkilökohtaisista tietokoneista, jotka alkoivat lännessä yleistyä 1970-luvun lopulla. Maa kärsi talousvaikeuksista eikä pystynyt vastaamaan kunnolla edes tutkimus- ja tuotantolaitosten kasvavaan tietotekniikan kysyntään. Naapurimaassa Bulgariassa tuotettiin vuodesta 1980 alkaen Apple II -yhteensopivia Pravetz-koneita opetuskäyttöön, mutta tuotanto jäi alkuvuosina vähäiseksi. Koska USA:n johtama CoCom oli asettanut vientikiellon korkealle teknologialle, ei tavallisille neuvostokansalaisille jäänyt ennen 1980-luvun jälkipuoliskoa muita vaihtoehtoja kuin itse rakentaminen ja salakuljetus.

Vuonna 1982 radioharrastelehti Radio (Радио) julkaisi kokoamisohjeet Mikro-80 (Микро-80) -nimiselle tieto-

koneelle. Kone käytti neuvostoliittolaisia K580-sarjan suorittimia, jotka olivat kopioita Intelin 8080:sta. Muistia oli 64 kilotavua ja ainoana näyttömoodina 64x32 merkin tekstitila. Koneessa oli yli 200 osaa, minkä vuoksi harrastelijat eivät koonneet niitä kuin muutamia. Osien löytäminen oli vaikeaa ja onnistui lähinnä mustan pörssin kautta. Mikro-80:sta julkaistiin yksinkertaistettu versio Radio 86RK (Радио 86РК) vuonna 1986. Koneessa oli enää 29 komponenttia. Prosessori ja tekstitila olivat samat kuin edeltäjässä, mutta muisti oli tyypitetty 16-32 kilotavuun. Koneesta tuli tietyissä harrastelijapiireissä suosittu, ja siitä tehtiin monia muunnelmia. Sille valmistettiin myös paljon lisäosia, esimerkiksi grafiikan esittämiseen kyennyt lisälaite.

Samana vuonna Mikro-80 -tietokoneen kanssa Neuvostoliiton radioministeriö julkaisi Apple II -kloonin nimeltään Agat-4. Siinä missä Bulgarian Apple-kloonit käyttivät suorittimena omaa 6502-klooniaan (CM630), neuvostomalleissa se korvattiin omintakeisella simulaativirityksellä. Vuoden 1984 valmistuserä jäi tyngäksi, ja vasta myöhemmät versiot Agat-7 ja Agat-9 levisivät massatuotantoon. Apple II -yhteensopivuus parani uusien mallien myötä. Agat oli suosittu opetuskäytössä, mutta harrastajien käsiin se ei suurissa määrin päätnyt.

Länsikoneita ei ollut kaikissa itäblokin maissa yhtä vaikeaa saada kuin Neuvostoliitossa. Pimeiltä markkinoilta hankittu Commodore 64 oli Puolassa, Jugoslaviassa ja Itä-Saksassa melko suosittu harrastekone, vaikka Neuvostoliitossa se oli täysin tuntematon. Ainoa Neuvostoliittoon virallisesti myyty kapitalistimaiden kotimikro lienee MSX, joita Yamaha myi opetustietokoneiksi.

Elektronika

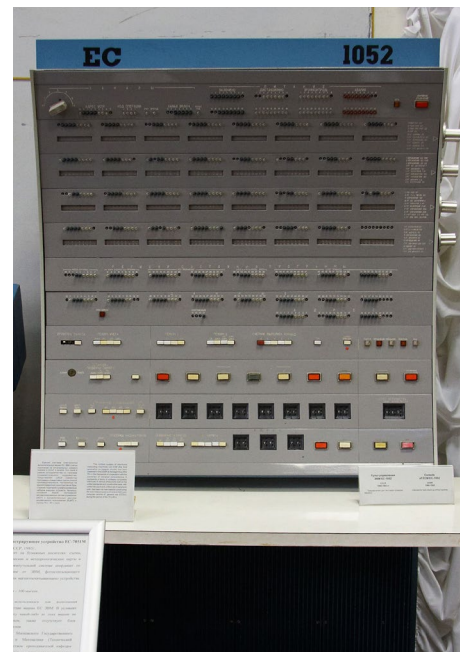
Siinä missä PDP-11-käskykantaan käytettiin länsimaissa lähinnä jääkaapin kokoisissa koneissa, oli se Neuvostoliitossa hyvin suosittu myös pienemmissä laitteissa ja sulautetuissa järjestelmissä, satelliitteja ja graafisia laskimia myöten. Pienet PDP-11-yhteensopivat koneet käyttivät samaa Elektronika-tuotenimeä (Электроника) kuin taskulaskimetkin. Elektronika-mallit 60, 85 ja DVK oli tarkoitettu pääte- ja työasemakäyttöön. Malli 60 tunnetaan koneena, jolle alkuperäinen Tetris ohjelmoitiin. Työasemasarjan pohjalta kehitettiin myös ensimmäinen virallinen neuvostokotimikro, Elektronika BK (БК). BK oli tarkoitettu



UKNC MS 0511



Elektronika MK-90



ES-1052:n hallintayksikkö



Nu Pogodi! -elektroniikkapeli

etupäässä koulukäyttöön, mutta se oli ensimmäinen tietokone jonka neuvostokansalaiset saivat virallisesti ostaa kotiin.

Ensimmäistä BK-sarjan konetta, BK-0010:aa, alettiin tuottaa 1985. Sen suoritin, K1801VM1, oli PDP-11-arkkitehtuurin mukaisesti 16-bittinen ja toimi 3 MHz:n kellotaajuudella. Keskusmuistia oli 32 kilotavua. Näytön resoluutioina toimivat kaksivärinen 512x256 ja nelivärinen 256x256 pikselin tila. Näissä moodeissa simuloitiin myös 64x25 ja 32x25 merkin tekstitiloja. Koneessa tuli mukana FOCAL-ohjelmointikieli, ja BASIC-tulkin sai lisävarusteena.

BK-0010:stä julkaistiin paranneltu versio BK-0010.01, jossa oli heikotuntu- maisen laudan sijasta kirjoituskonenäp- päimistö. Myös BASIC-tulkki oli valmiina ROM-muistilla. Kouluja varten julkaistiin BK-0010Š, johon kuului monitori ja verkottamismahdollisuus. Vuonna 1989 jul- kistettu BK-0011M nosti suorittimen kel- lotaajuuden 4 MHz:iin ja keskusmuistin 128 kilotavuun. Lisäksi graafisia ominai- suuksia paranneltiin: paletti ei ollut enää vain musta, punainen, vihreä ja sininen, vaan tarjolla oli 15 muutakin neliväripa- lettia.

1990-luvulla BK-0011M:n käyttäjät varustivat koneensa AY-äänipiirillä sekä

levykeasema- ja kiintolevyohjaimilla. Spectrum-demoskenen nousun myötä alkoi ilmestyä myös BK-demoja.

Muita koneita

Vuonna 1987 julkistettua, kouluuihin tar- koitettua UKNC (YKHII) -konetta voisi kutsua neuvosto-Amigaksi niin ulkokuo- rensa kuin sisuskalujensakin puolesta. UKNC:ssä oli kaksi PDP-11 -yhteenso- pivaa K1801VM2-prosessoria, joista toi- nen toimi grafiikka- ja apuprosessorina. Pääprosessorin kellotaajuus oli 8 MHz, keskusmuistia oli 192 kilotavua ja gra- fiikkatilassa 8 väriä 640x288 pikselin tarkkuudella.

“Neuvosto-Amigan” ohi multimedia- ominaisuuksissa menee samana vuonna kehitetty 8-bittinen kotimikro, Vektor- 06C (Бекроп-06И). Kun muut neuvosto- koneet tyytyivät piippausääniin, Vekto- rista löytyi äänikäyttöön pyhitetty Intelin 8253-laskuripiirin klooni, joka tarjosi kolme kanttiaaltokanavaa. Kokonaisvä- ripaletti oli peräti 256 väriä, joista oli 256x256 pikselin grafiikkatilassa kerralla käytössä 16. Pullonkaulaksi jäi suoritti- mena oleva 8080-klooni, joka ei kolmen megahertsin kellotaajuudellaan jaksa- nut piirtää näytölle kovin nopeasti. Kes- kusmuistia oli 64 kilotavua, josta puolet

näyttömuistia. Laite oli varsin suosittu, ja sille tehtiin paljon pelejä.

Neuvostotietotekniikasta puhutta- essa ei sovi unohtaa taskulaskimia, joista muutamaa myytiin Teboil-huoltamoilla Suomessakin. Laskimia tuotettiin luke- mattomia malleja yksinkertaisista neli- laskimista ohjelmoitaviin graafisiin laski- miin, joiden kermää edustaa huippukallis Elektronika MK-90 -taskutietokone. MK- 90:ssa oli 160x64 pikselin nestekide- näyttö, BASIC-tulkki ja PDP-11 -yhteen- sopiva suoritin. Kellotaajuus jouduttiin pitämään sen verran alhaisena että laite oli hidas. Laskimen mukana tuli tyhjän tallennusmodulin lisäksi ROM-moduli, josta löytyi mm. Tetris, Pac-Man-klooni ja šakkipeli.

Sosialistimaissa tuotettiin myös pelikäyttöön tarkoitettua tietotekniik- kaa. Neuvostojärjestelmä suolsi sekä taskuelektroniikkapelejä, jotka olivat enimmäkseen suoria kopioita Nintendon Game&Watch -peleistä, että kolikkope- lejä. Useimpien kolikkopelien tekniset puitteet olivat jättimäisine pikseleineen vaatimattomat, mutta muutamissa yllät- tävän edistyneet. TIA-MC-1 -nimisestä laitteistosta löytyi mm. 256 väriä ja spri- tet.



Bulgarialaiset Apple-kloonit nähtävästi toimivat pereslavlilaisessa koulussa vuonna 1985, mutta toimiiko keskuslämmitys? Kuva: Pereslavskaja ja pedelja -lehti

Kloonatut kasibittiset

Vaikka Neuvostoliitossa ja itäblokin maissa tuotettiin useita omaperäisiä koneita, päätyivät kaikkein suosituimmiksi kotimikroiksi brittiläisen ZX Spectrumin kloonit. Spectrum oli ihan-teellinen kone kloonattavaksi. Se oli pienen kokonsa ansiosta helppo salakuljetettava, ja tekninen yksinkertaisuus teki sen toiminnan selvittämisen ja uudelleentoteuttamisen helpoksi. Ensimmäinen itäspectrum rakennettiin 1985, ja 1990-luvun alkupuolelle tullessa oli miltei jokaisessa suuremmassa neuvostokaupungissa Spectrum-valmistajia. Esikuviaan kehittyneemmiksi edenneistä Spectrum-klooneista kerrotaan enemmän tämän numeron klassikkokoneita käsittelevässä artikkelissa.

Vaikka Spectrum oli ylivoimaisesti suosituin kasibittinen kone sosialistisissa maissa, myös muita kasibittisiä kloonattiin. Aiemmin mainitut Agat ja Pravetz olivat Apple II -klooneja, paitsi Pravetz-malli 8D, joka oli Oric-klooni. Jugoslaviassa puolestaan julkistettiin jo vuonna 1983 Galaksija-niminen ZX81-klooni, jossa käytettiin lännestä tuotuja Z80-suorittimia. Kiinassa on kloonattu 1980-luvun lopulta alkaen NES-pelikonsoleja, jota päätyi myös Neuvostoliiton ja Itä-Euroopan markkinoille.

Harrastuskulttuuri

Neuvostoliiton tietokoneteollisuudelle oli ominaista se, ettei erillisen ohjelmistotuotannon tarvetta huomioitu viisivuotissuunnitelmissa. Niin harrastus- kuin ammattikäyttäjätkin joutuivat kirjoittamaan ohjelmansa alusta lähtien itse ja tekemään erilaisia laitteistomuunnoksia. Tämä synnytti voimakasta itse tekemisen kulttuuria ja harmaata taloutta, joka vertautuu länsimaisten hakkerikulttuurien epäkaupallisuuteen. Samalla, kun ohjelmistotuotanto oli kapitalistimaissa isoa bisnestä ja vapaata kopiointia tapahtui lähinnä yliopistoissa ja harrastajapiireissä, oli kuvio sosialistimaissa päinvastainen. Ohjelmistojen vapaa kopiointi oli yleisesti hyväksytty normi, ja rahaa kopioista pyydettiin lähinnä harrastajapiireissä ja pimeillä markkinoilla.

Etenkin BK-koneen peleihin ja ohjelmiin kuuluu tyypillisesti alkuruutu, jota voi verrata länsipiraattien crack-introihiin. Siinä ohjelman alkuperäinen tekijä tai jälleenmyyjä esittelee itsensä, mainostaa muita myynnissä olevia pelejään ja antaa yhteydenottoja varten puhelinnumerosa. Epävirallista ohjelmistotuota-

tantoa oli BK:lle hyvin paljon. Harrastajat käänsivät sille sekä länsikoneiden että neuvostomikrojen ohjelmia ja tuottivat paljon pelejä.

Ensimmäiset sosialistimaiden demot tehtiin Puolassa ja Tšekkoslovakiassa Spectrum-koneille, mistä ilmiö siirtyi ohjelmakopioverkostojen kautta Neuvostoliittoon. Itäblokin demontekijöillä ei ollut alkuvaiheessa mitään tietoa länsimaisen demoskenen olemassaolosta. He ryhtyivät puuhaan täysin omista lähtökohdistaan käsin. Neuvostoharrastajat kohtasivat länsimaisen demoskenen ensimmäistä kertaa Amiga-demojen kautta, vaikkei juuri kenelläkään ollut Amigaa. Demot levisivät VHS-kasetteina kopion kopioina, ja Spectrum-harrastajat ottivat niistä vaikutteita demoihinsa. Muutamissa demoissa saatettiin todeta "Amiga rules!", vaikka tekijät eivät koskaan olleet nähneet aitoa Amigaa..

Lännen ja idän demoskenet löysivät toisensa vasta vuosia Neuvostoliiton hajoamisen jälkeen, 1990-luvun jälkipuoliskolla, jolloin "ex-USSR"-demokulttuuri vakiintui ja suurempia demotapahtumia alettiin järjestää säännöllisesti. Itädemoskenen pääkoneeksi päättyi tällöin Spectrum-klooni Pentagon 128, joka toimii edelleen venäläisten Spectrum-demojen standardina.

Yhteenveto

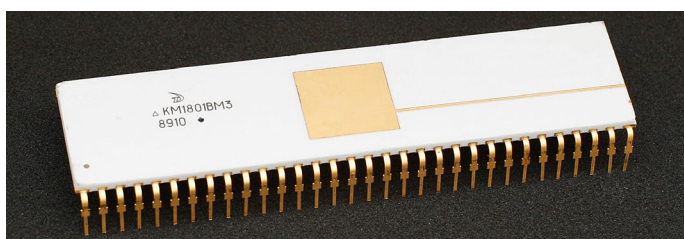
Neuvostoaajan kotimikroille on yhteistä se, että suorittimet olivat kopioita länsimaisista mutta muu tekniikka omintakeista. Tekniikka oli ajankohdasta ja laskutavasta riippuen noin 5-10 vuotta länsikoneita jäljessä. Varhaiselle 1970-luvun mikrotietokonekulttuurille ominainen itse rakentaminen oli Neuvostoliitossa yleistä vielä 1980-luvun jälkipuoliskolla. Ohjelmistoja ja lisälaitteita ei ole ollut myynnissä, ja harrastelijat tekivät ne itse. Neuvostoaikaa leimasi verevä tee-se-itse -kulttuuri, jonka perintö on yhä voimissaan!



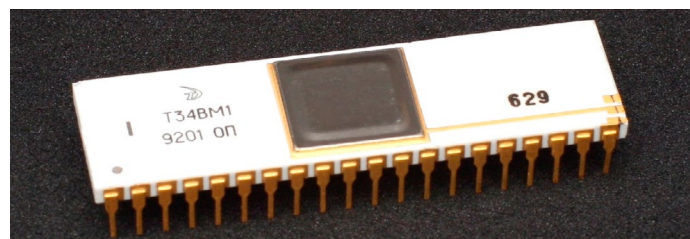
K1801BM1



KP580BM80 (= Intel i8080)



KM1801BM3 (KM1801WM3)



T34BM1 (= Zilog Z80)

Bytebeat-musiikkia

Ville-Matias Heikkilä

Bytebeatit ovat lyhyitä koodinpätkiä, jotka tuottavat chip-musiikin tapaista ääntä, kun ne sijoitetaan äänipuskuria täyttävään silmukkaan. Allaolevia kaavoja voi kokeilla vaikkapa nettisivullamme skrolli.fi/bytebeat.

```
t&t>>8
t>>6&1?t>>5:-t>>4
t*(t>>9&t>>13)&16
(t*5&t>>7)|(t*3&t>>10)
(t&t%255)-(t*3&t>>13&t>>6)
t*((t>>9)^((t>>9)-1)^1)%13)
t*(51864>>(t>>9&14)&15)|t>>8
t*((t>>12)|(t>>8)&(63&(t>>4)))
(t>>7|t|t>>6)*10+4*(t&t>>13|t>>6)
(t*9&t>>4|t*5&t>>7|t*3&t/1024)-1
((t*(t>>8|t>>9)&46&t>>8))^ (t&t>>13|t>>6)
(t*((3+(1^t>>10&5))*(5+(3&t>>14))))>>(t>>8&3)
```

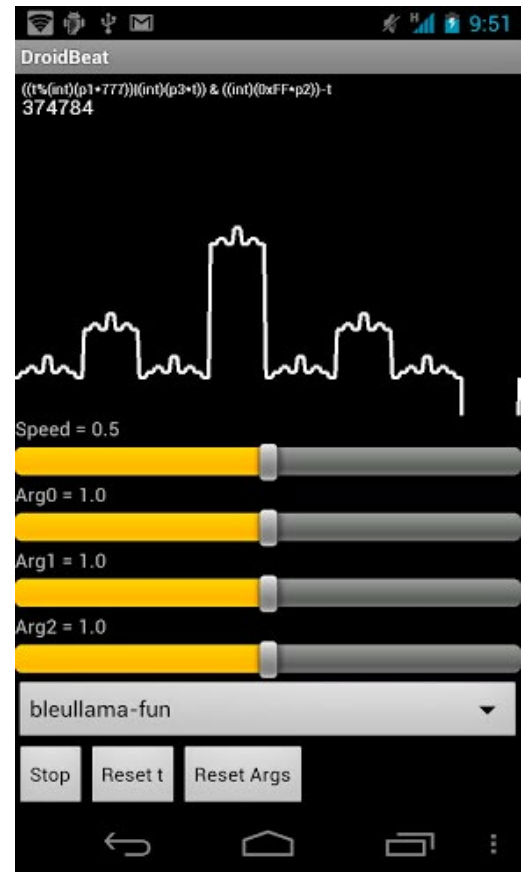
Unix-komentoriviltä bytebeateja voi kokeilla C-ohjelmina. Seuraava komentorivi kääntää kaavaa `t&t>>8` soittavan ohjelman ja ajaa sen:

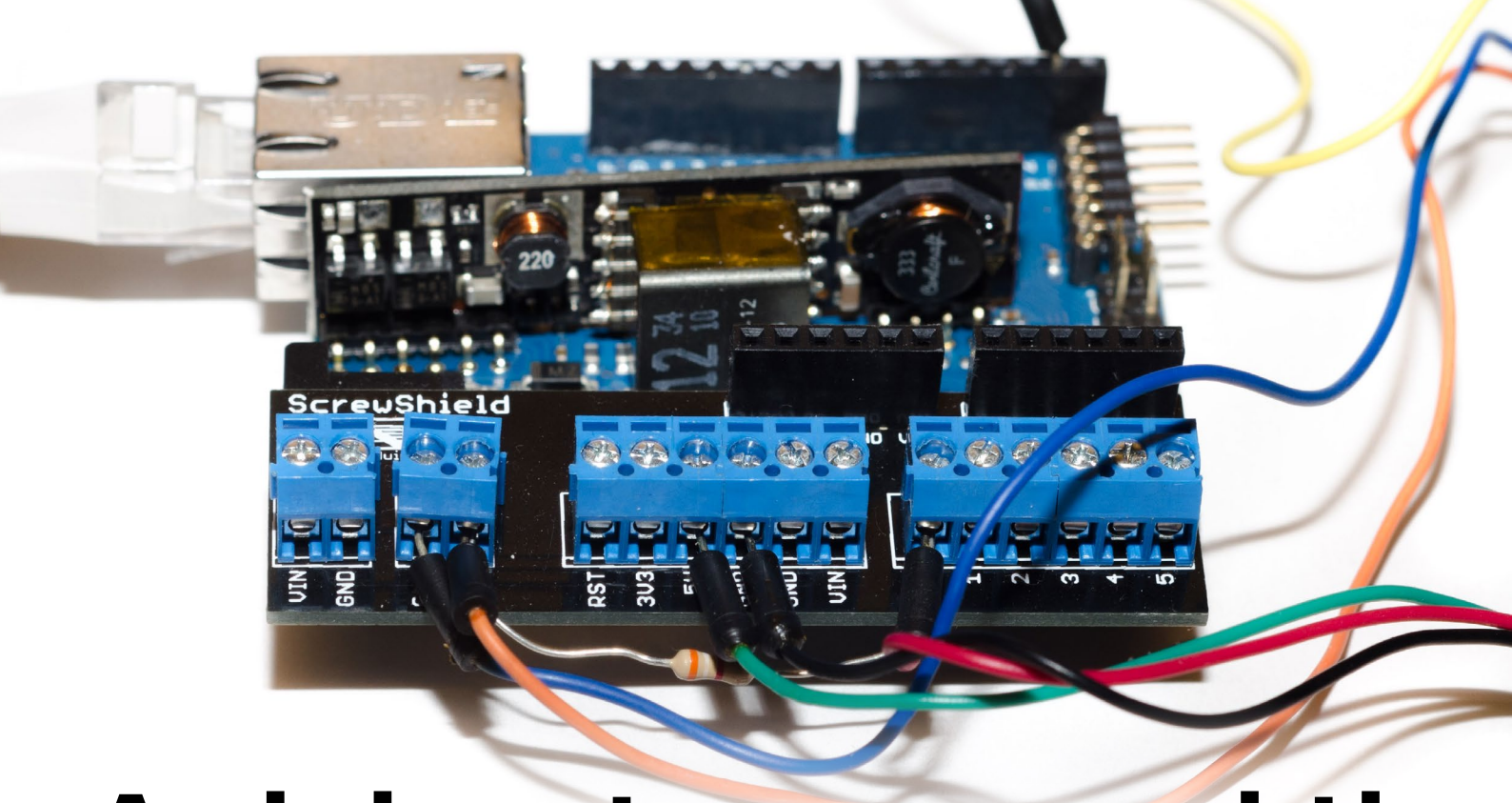
```
echo 'main(t){for(;;t++)putchar(t&t>>8);}'|gcc -x c -&&./a.out|play
```

Loppuosan `|play` ohjaa tulostuksen ALSA-äänipalvelulle. Sen voi joutua eri järjestelmissä muuttamaan esimerkiksi muotoon

```
|pacat --format=u8 --rate=8000 --channels=1 tai
>/dev/dsp
```

Bytebeat-musiikin tekoon sopivia ohjelmia saa myös mobiililaitteille. iOS:lle tällaisia ohjelmia ovat BitWiz Audio Synth ja GlitchMachine, Androidille taas DroidBeat. Näistä GlitchMachine käyttää C-syntaksin sijaan Forth-tyylistä pinokieltä.





Arduinosta saunavahti

Teksti ja kuvat: Anssi Kolehmainen
*Ulkona on kylmää, hyistä ja pimeää, mutta sauna on onneksi lämpiämässä. Kohta pääsee löylyyn! Mutta onko-
 han kiuas jo lämmin?*

Sen sijaan, että vaeltaisimme saunalle asti tarkastamaan tilanteen, ratkaisemme ongelman tekniikalla. Rakennamme laitteiston, joka kertoo saunan lämpötilan reaaliajassa suoraan taskupuhelimeen.

Saunan lämpötilasta raportoiva mittarimme koostuu yleiskäyttöisistä osista, joilla voi tehdä kaikenlaista kukkien kastelujärjestelmistä ovisummereihin ja CNC-jyrsinten ohjaimiin. Siinä missä kaupasta ostettu lämpömittari pystyy ainoastaan mittaamaan lämpötiloja, tämä laite on muokattavissa vaikkapa soittamaan summeria, kun sauna on valmis tai kun lämpötila lähtee uhkaavaan laskuun.

Esimerkkilaitteistoa voi käyttää muidenkin lämpötilojen mittaukseen pelkän saunan seurannan sijaan. Kaikenlaista elektroniikkaa rakentaessa voi laitteiden mahdollisen ylikuumenemisen selvittää laittamalla lämpötilasensorit laatikon sisään ja katsomalla tulokset ruudulta. Sensorit voi laittaa myös esimerkiksi jääkaappiin, pakastimeen tai kellarikomeroon tarkkailemaan ruokien säilyvyyttä.

Komponentit

Itse lämpötilamittaus on toteutettu DS18B20-piirillä. Se kuuluu Dallas Semi-

conductorin kehittämään 1-Wire -tuotepereeseen, josta löytyy monenlaisia pieniä sensoripiirejä. Tyypillisimpiä ovat lämpötila- ja kosteussensorit, laskurit ja jännitemittarit. Sensorit ovat digitaalisia ja kommunikoivat 1-Wire -protokollan avulla. Rautasuunnittelua tämä helpottaa huomattavasti, sillä samoihin piuhoihin voi kytkeä useita sensoreita niiden häiritsemättä toisiaan. Niitä voi kytkeä kiinni lennosta, ja koska jokaisella on oma uniikki sarjanumero, ne eivät sekoitu keskenään.

Digitaalinen kommunikaatio pienillä nopeuksilla antaa myös paljon valinnanvaraa kaapeloinnin suhteen. Itse olen päättänyt käyttämään tavallista audiopiuhaa ja 3.5mm plugiliittimiä. Etuna tässä on se, että jatkojohtoja saa lähimmästä K-kaupasta. Toinen hyvä asia on stereojakajien käyttömahdollisuus. Lisäsensorin kytkeminen vaatii normaalin audiojakaajan lisäämisen, jonka jälkeen uuden sensorin voi kytkeä kiinni. 1-Wire -tuotteiden kanssa käytetään tyypillisesti lankapuhelimista tuttua RJ11-liitintä, mutta sellaisen tekeminen vaatii puristuspihtejä, eikä jakapaloja enää nykyisin löydy kaupasta. Mikään ei tietenkään estä käyttämästä vaikka vanhaa verkkopiuhaa ja kolvaamasta liittimiä suoraan kiinni sensoriin. Paras ratkaisu riippuu käyttökohteesta. Valmistajasta riippuen 1-Wiren pitäisi toimia jopa satojen metrien kaapeleilla, mutta itse olen huomannut ongelmia jo kymmenen metrin matkalla.

Seuraava vaihe järjestelmässä on tiedon kerääminen sensoreilta. Se on hoidettu Arduino Ethernet -mikrokontrollerilla, joka kyselee jatkuvasti sensoreilta uusia lämpötilatietoja ja lähettää ne eteenpäin. Arduino-tuotepere on hyvin tuotetettu, aloittelijaystävällinen elektroniikan häkkäysalusta. Arduinoilla voi tehdä "ihan mitä tahansa", esimerkiksi tällaisen lämpötilamittarin tai vaikkapa ovisummerin, valo-ohjaimen tai puhuvan leivänpaahtimen. Tarvitaan vain kourallinen kivoja elektroniikan komponentteja, vähän koodaustaitoa ja intoa kokeilla asioita.

Käyttövirran koko järjestelmä saa verkkokaapelia pitkin Power over Ethernetillä. Tässä projektissa se selvästi helpottaa johdotusta, koska tarvitsee vetää vain yksi kaapeli Arduinolle. Toisessa päässä on PoE-injektori, joka kytketään normaaliin seinäsähköön ja verkkopiuh-



DS18B20 -lämpömittari on melko pieni.

hoihin. Tehoa saa välitettyä enintään 15W, joka riittää moneen projektiin aivan loistavasti. Kannattaa huomata, että on olemassa myös epästandardeja PoE-ratkaisuja, joita myydään melkein samalla nimellä.

Seurannan helpottamiseksi Arduinoon on lisätty myös LCD-näyttö, joka kertoo senhetkiset lämpötilat. Näyttö itsessään on vakiolaite ja liitetty Sparkfun SerLCD -adapterin avulla, joka pudottaa vaadittavien ohjaussignaalien määrän viidestä yhteen. Ohjelmoinnin kannalta SerLCD:n käytöllä ei ole merkitystä, mutta se helpottaa johdotusta.

Lämpötilat koneelle

Kun sensorien data on saatu numeroiksi, se pitää vielä ottaa talteen ja saada näkyviin kännykkään. Kuten Ethernet-termistä saattoi arvata, Arduino lähettää mittaustulokset verkkoon, josta PC-kone (tai vaikka Raspberry Pi) poimii ne talteen. Laiskana koodarina laitoin Arduinon lähettämään tiedot broadcastina koko verkkoon, jolloin ei ole väliä mitä IP-osoitteita käytetään.

Vastaanottopuolelle koodasin muutaman rivin Perl-taian, joka kuuntelee verkkoa ja kirjoittaa tulokset ylös. Sama onnistuisi melkein millä tahansa ohjelmointikielellä.

Tietojen varastointiin ei tarvitse välttämättä koodata omaa softaa, vaan voi käyttää rrdtoolia. Sillä luodaan aluksi halutun kokoinen arkisto tarvittavista mittareista, esimerkiksi katto- ja istumatasojen lämmöt yhden vuorokauden ajalta. Tämän jälkeen tietoja päivitetään sitä mukaa kun niitä syntyy. Rrdtool huolehtii myös aikaleimojen keräämisestä, joten ei haittaa vaikka järjestelmä olisi välillä pois päältä.

Lopuksi koneen tai kännykän ruudulle saadaan piirrettyä lämpötilakäyrä rrdtoolia käyttäen. Kokemukseni mukaan on helppointa tehdä tämän www-palvelimen kautta muutaman rivin PHP-skriptillä. Tällöin samaa sivua voi käyttää omalta koneelta, läppärltä, kännykältä tai vaikka kummin-kaiman tabletilla. Kun projektin jälkeinen tylsyys yllättää, voi sivulle koodailla lisää animaatioita ja hienouksia.

<http://arduino.cc/>

Arduino-projektin kotisivut

<http://www.sparkfun.com/>

Suuri yhdysvaltalainen elektroniikkakauppa

<http://paeae.com/>

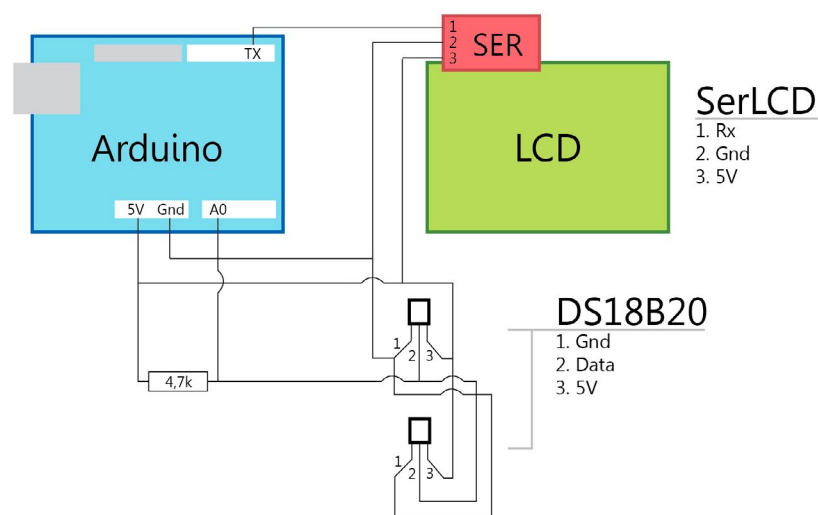
Kotimainen Arduino-/elektroniikkakauppa

<http://partco.biz/>

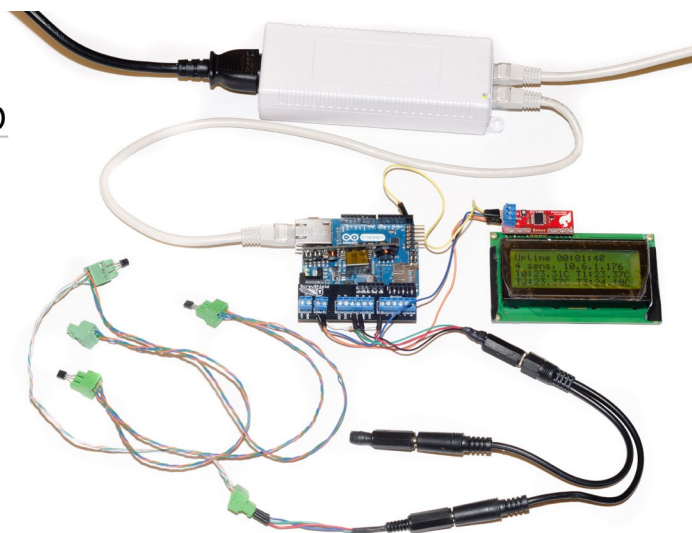
Suuri suomalainen elektroniikkakauppa

<http://status.futurice.com/page/sauna>

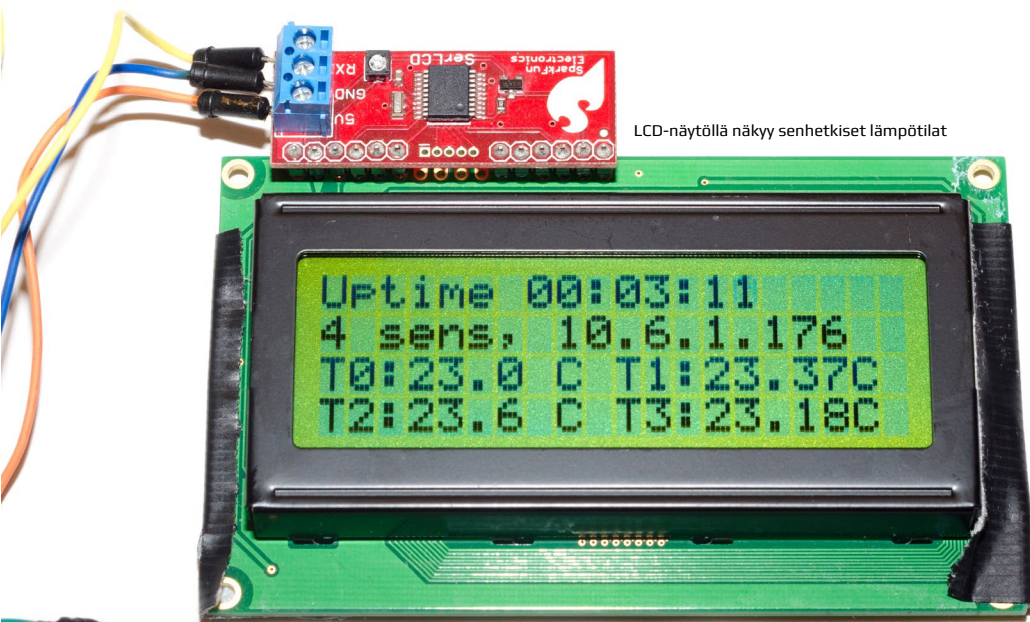
Erilainen toteutus saunalämpömittarista



Laitteiston kytkentäkaavio



Koko järjestelmä kasassa. Yksi lämpömittari 3,5 mm plugissa, loput kytkettynä toiseen kaapeliviritykseen. Virta tulee PoE-injektorin kautta.



LCD-näytöllä näkyy senhetkiset lämpötilat

Päihitä Thinkpadin suojat

Kun työkone lukittuu, leikki on kaukana. Näin purat lukitun Thinkpadin suojaukset.

Risto Järvinen

Tietokoneiden käyttörajaukset ovat suoja luvaton käyttöä vastaan ja parantavat tietoturva. Salasanat kuitenkin hukkuvat, ja lukkiutuneet koneet voivat päätyä elektroniikkaromuksi. Suojaukset perustuvat yleensä salasanaan, mutta yritysmaailmassa myös älykortit ja sormenjälkitunnistimet ovat yleisiä. Nämä toteutetaan usein käyttöjärjestelmän tasolla, jolloin koneen tyhjentäminen tai kovalevyn vaihtaminen johtaa siihen, että kone on puhdas ja valmis uusiokäyttöön. Osa laitevalmistajista on laajentanut suojausmenetelmiä. Suurin yksittäinen samanlaisia laajennuksia hyödyntävä ryhmä on IBM:n/Lenovan kannettavat tietokoneet.

Artikkelissa tehdään katsaus PC-tietokoneiden suojausmenetelmiin ja niiden purkamiseen. Syvennymme Thinkpad-kannettavien suojauksiin, sillä niiden tahaton pysyvästi lukitseminen on yleinen ongelma. Tämä tulee usein vastaan sellaisissa yrityksissä, joissa halutaan käyttää kaikkia mahdollisia suojia yrityssalaisuuksien suojaamiseen. Esittelemme purkumekanismien toiminnan BusPirate-työkalua käyttäen.

PC:n rakenne ja BIOS

PC-tietokoneiden käynnistäminen perustuu BIOSiksi kutsuttuun ohjelmaan. BIOS (Basic Input/Output System) tarkoittaa käyttöjärjestelmän perusosaa, joka alustaa tietokoneen laitteiston ja tarjoaa rajapinnat sen käyt-

tämiseen. DOS-käyttöjärjestelmät käyttivät vielä paljon BIOS-palveluja, mutta nykyiset käyttöjärjestelmät korvaavat ne omilla, tehokkaammilla rajapinnoillaan. Nykyroolissaan BIOS säätää joitain tietokoneen asetuksia ja aloittaa varsinaisen käyttöjärjestelmän käynnistystyksen.

PC-BIOS on toteutettu FLASH-piirillä, joka sisältää BIOS-firmwarea, ja paristovarmistetulla "CMOS-muistilla", jossa säilytetään BIOS-asetukset ja kellonaika. CMOS-muisti toteutettiin alun perin MC146818-reaaliaikakellopiirillä, jossa oli 64 tavua RAM-muistia. FLASH-piirit ovat yhä erillisiä osia emolevyllä. Suurin niistä on noin peukalonpään kokoinen 28-pinninen rinnakkais-FLASH-piiri ja pienin taas 8-pinninen, nuppineulan päään kokoinen SPI-FLASH-pintaliitospiiri. CMOS-muisti on nykyään yleensä osa piirisarjaa.

Nimi "CMOS-muisti" tarttui käyttöön, kun paristovarmistettuja reaaliaikakellopiirejä haluttiin valmistaa energiatehokkaammalla tekniikalla (Complementary Metal Oxide Semiconductor) kuin millä silloisia piirejä yleensä valmistettiin (NMOS, N-type Metal Oxide Semiconductor). Alkuperäiset IBM PC -koneet käyttivät muistista nimeä "CMOS RAM", ja termiä on käytetty siitä lähtien, vaikka

nykyään liki kaikki mikropiirit käyttävät CMOS-valmistustekniikkaa.

CMOS-muistilla on paristovarmennus, joka on liki poikkeuksetta toteutettu CR2032-litiumparistolla. Osa emolevyistä tarjoaa suoraan tavan tyhjentää CMOS-muistin kytkemällä jumpperin muistintyhjennysasentoon. Kuvassa 1 Dell Optiplex 745 -tietokoneen emolevyiltä löytyvä "PSWD"-jumpperi, jonka irrottaminen poistaa salasanasuojaus.

Jos tällaista jumpperia ei löydy, yleisin tapa poistaa salasanat on CMOS-muistin tyhjentäminen irrottamalla paristo. Kannettavien tietokoneiden tapauksessa kotelon avaaminen paristoon käsiksi pääsemiseksi voi olla vaivalloista.

CMOS-muistiin pääsee käsiksi myös käyttöjärjestelmän sisältä, ja tähän on olemassa erityistyökaluja. CmosPwd voi lukea CMOS-muistin, ottaa siitä kopion, palauttaa kopion, tyhjentää muistin tai jopa yrittää päätellä, mikä BIOS-salasanana on. Työkalun käyttöön tarvitaan ylläpitäjän oikeudet, koska sillä ohitetaan tietokoneen turvamekanismeja. Salasanan tallentamiseen CMOS-muistiin ei ole vakioituja käytäntöjä, joten salasanan päättely ei useinkaan toimi. Monet BIOS-toteutukset eivät tallenna salasanoja CMOS-muistiin, joten ne ovat lähtökohdaisesti immuuneja tälle menetelmälle.



Kehittyneemmät menetelmät

Kannettavat tietokoneet ovat liikkuvan luonteensa takia alttiimpia varkauksille ja katoamisille. Varkauksien estämiseksi on muutamia menetelmiä:

- Trusted Platform Module (TPM)
- kovalevysuojaukset
- seurantapalvelut

Trusted Platform Module on apuprosessori, joka toimii yleensä muiden menetelmien tukena. Yksi sen olennaisin toimintamenetelmä on salasanojen tallentaminen. Tätä käytetään usein kovalevysalauksen salasanojen säilömiseen. Yksi TPM-moduulien alkuperäisistä käyttötarkoituksista oli, osana Trusted-Computing -järjestelmää, estää lisensoimattomien ohjelmistojen suorittaminen. Trusted-Computing -järjestelmän vastustus hidasti TPM-moduulien yleistymistä.

Seurantapalveluita on kahta eri tyyppiä: käyttöjärjestelmästä riippumatta taustalla toimivia ja ohjelmallisia, käyttöjärjestelmän alle asennettuja. Esimerkkinä ensimmäisistä on Intel AMT -teknologia, jolla tietokone voi raportoida tilastaan ja sijainnistaan verkkoon. Sillä voidaan myös lähettää tietokoneelle hallintaohjeita. CompuTrace on kaupallinen seurantapalvelu, joka voidaan integroida BIOSiin.

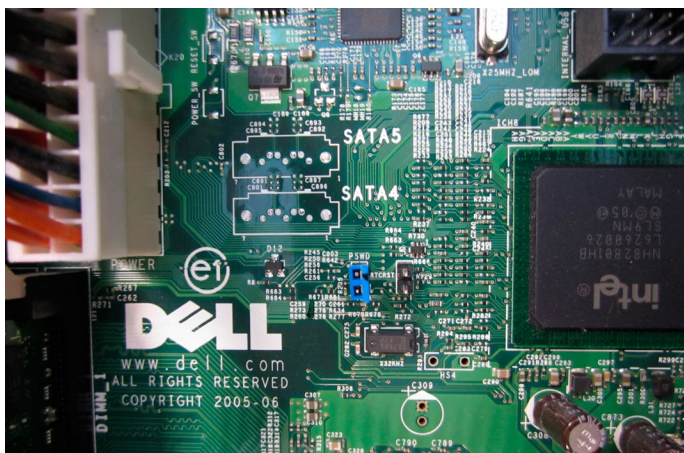
Ohjelmalliset seurantapalvelut perustuvat käyttöjärjestelmän mukana käynnistyviin ohjelmiin, jotka pyrkivät muodostamaan internet-yhteyden ja lähettämään tunnistettavia tietoja seurantapalveluun. Yleisiä lähetettyjä tietoja ovat nykyinen verkko-osoite, kuvakaappaukset ja jopa kaapatut näppäimistön painallukset. Tietoja voi käyttää koneen paikantamiseen. Ohjelmallisten seurantapalvelujen heikkous on, että muiden ohjelmistojen lailla ne voidaan poistaa esimerkiksi tyhjentämällä kovalevy.

Kovalevysuojaukset perustuvat ATA-komentoihin, joilla kovalevyä voidaan käskä vaatimaan salasanaa sisäänkäsyn hallitsemiseksi. Osa kovalevyistä myös salaa tallennettavan datan, jolloin kovalevyjen elektroniikkaa vaihtamalla tai suojaukset nollaamalla saattaa levyn saada auki. Tallennettuun dataan ei kuitenkaan silloin pääse käsiksi.

Kovalevysuojauksien purkuun on kaksi käytännöllistä vaihtoehtoa:

1. Suojauksen purkaminen oikealla salasanalla
2. Kovalevyn turvatyhjennys saattaa purkaa suojauksen

Jotta salatulua kovalevyltä voisi bootata, BIOSin tulee tukea kovalevysalasanvoja. Äärimmäisessä hätässä kovalevyn elektroniikat voi yrittää vaihtaa toisesta täsmälleen saman mallin



Kuva 1: Salasanan tyhjentämiseen tarkoitettu jumpperi tietokoneen emolevyltä.

levystä. Tämä on kuitenkin riskialtis ja todennäköisesti tuhoon tuomittu operaatio, eikä vaihto ole mahdollinen SSD-kovalevyjen tapauksessa.

UEFI – tulevaisuuden kasvat

Unified Extensible Firmware Interface on Intelin kehittämä korvike BIOSille. Alkuperäinen motivaatio BIOSin korvaamiseen oli tarve saada BIOS-ratkaisu Intelin uudelle, yhteensopimattomalle Itanium-prosessorille. Toisin kuin BIOS, joka on prosessorikohtainen ohjelmisto, UEFI:n mallin mukaan laitevalmistajat tekisivät ajurit laitteilleen tavukoodimuodossa, jota UEFI:ssä oleva tulkki voisi suorittaa.

Kehitys lähti liikkeelle Inteliltä, ja useimmat valmistajat käyttävät UEFI:n toteutuksena pohjana Intelin julkaisemaa referenssitoteutusta. Itanium-prosessorien ja 32-bittisten prosessorien marginalisoiduttua UEFI-toteutukset ovat rajoittuneet x86-64. BIOSiin verrattuna UEFI-toteutukset ovat melko hiomattomia ja niistä on löydetty monia bugeja. UEFI-tason ongelmat estävät yleensä tietokonetta käynnistymästä ja ovat siten huomattavasti tavallista ohjelmistovikaa vakavampia.

Tietokoneiden suojauksen kannalta UEFI on yhä samanlainen musta laatikko kuin BIOS. UEFI-standardi määrittelee minimimäärän pysyvää muistia, joka UEFI-toteutusten pitää tarjota. Käytännössä tämä tarkoittaa, että tietokoneisiin on integroitava isompi muistipiiri. BIOSin mukana katoaa myös tarve tarjota CMOS-muistia yhteensopivuuden takaamiseksi.

Windows 8 asetti vaatimuksen sisällyttää SecureBoot-mekanismi UEFI-toteutuksiin. Windows RT -pohjaisissa tietokoneissa (lähinnä ARM-pohjaisia tabletteja) SecureBoot tulee olla aina kytkettynä päälle. PC-koneissa SecureBootin saa toteuttaa niin, että sen voi kytkeä pois päältä. SecureBoot ei sinänsä ole tietoturvaominaisuus, vaan se rajaa ohjelmistoja, joita tietokone saa suorittaa. Jos ohjelmisto on hyväksytty eli esimerkiksi sisältää Microsoftin digitaalisen allekirjoituksen, tietokone voi suorittaa sen. Allekirjoittamattomia ohjelmistoja ei suoriteta. Käyttäjää tuskin lohduttaa, että kannettavan varastanut pitkäkyntinen ei voi ajaa sillä FreeBSD:tä, sillä yksityiset tiedostot voi edelleen kopioida talteen ja asentaa puhtaan Windows 8:n ennen laitteen myymistä eteenpäin.

IBM/Lenovo Thinkpad -suojaus

IBM ja sen PC-yksikön ostanut Lenovo ovat rakentaneet Thinkpad-kannettaviin tavallista monimutkaisempia suojausmekanismeja. Thinkpadeja markkinoidaan ensisijaisesti yrityskäyttöön, joten suojauksille on kysyntää. Monissa Thinkpad-malleissa on sormenjälkilukija ja tuki älykortin kytkemiselle.

Thinkpadin BIOS käyttää CMOS-muistin lisäksi pientä EEPROM-muistia ja valinnaisesti TPM-piiriä salasanojen tallentamiseen. Tarkat menetelmät riippuvat tietokoneen mallista ja käytetyistä BIOS-asetuksista. EEPROM-muistin tyyppi vaihtelee, mutta se on yleensä yhteensopiva Atmel 24RF08 -piirin kanssa. Muisti on kytketty I²C-yhteensopivan SMBus-väylän kautta piirisarjaan. I²C-väylätekniikka perustuu kahteen avoin kollektori -tilassa toimivaan linjaan, mikä sallii väylälle kytketymisen.

Seuraavaksi käytämme Dangerous Prototypesin kehittämää Bus Pirate -työkalua, jonka avulla terminaaliohjelmistolla voi helposti keskustella I²C-väylän kanssa. Lukemalla muistipiirin sisältö I²C-väylältä voidaan yrittää tulkita tallennettu salasana. Jos tulkinta ei onnistu, voidaan salasanan sisältö ylikirjoittaa muistista, jolloin BIOS päättelee, ettei salasanaa ole asetettu.

BusPirate osaa myös passiivisesti kuunnella I²C-väylän liikennettä. Terminaali ohjelmistoksi sopii Unix-sukuisilla Minicom ja Windowsilla PuTTY. Sarjaportin moodiksi asetetaan 8N1, 115200bps ja ei kättelyä.

Kannettavan tietokoneen rakenne ja käytetyn suojauksen yksityiskohdat vaihtelevat koneen mallin ja iän mukaan. Tässä käymme läpi esimerkin vuoksi Lenovo Thinkpad X61s -tietokoneen purkamisen. Jos vastaavaan projektiin haluaa ryhtyä, useimmilta valmistajilta löytyy verkosta huolto-oppaita, joissa neuvotaan kannettavan purkaminen osiin. Lisäksi Thinkpadien avaamisesta on tehty pienimuotoinen bisnes. Avaamispalvelun tarjoajat näyttävät, mistä kohdasta eri mallien emolevyihin tulisi kytkeytyä. Vältämme kuitenkin hämäräbisneksen tuemista ja kerromme, mihin käytetty menetelmä perustuu.

Oletetaan lähtötilanteeksi, että meillä on kannettava tietokone, jolle on asetettu PowerOn- ja Setup-salasanat. Kone kysyy käynnistyksessä salasanaa eikä suostu tekemään mitään, ellei salasanaa syötetä onnistuneesti. Tällaisessa tilanteessa salasanan poistamisesta ei synny haittaa. Jos myös kovalevy-suojaus olisi päällä, jouduttaisiin todennäköisesti hankkimaan uusi kovalevy lukitun tilalle.

Kuvassa 3 on esitelty kytkennän kaavakuva. BusPirate-laite kytketään mukaan järjestelmän I²C/SMBus-väylään. Vihreällä loholla esitetty I²C-muistipiiri toimii orjalaitteena, joka vain vastaa muiden esittämiin pyyntöihin. Punaisilla lohkoilla merkityt BusPirate ja järjestelmän oma ohjain toimivat väylällä mestareina, jotka lähettävät väylälle pyyntöjä. Jos useampi mestari toimii väylällä yhtä aikaa, voi näiden viestintä sotkeutua. Onneksi tietokone käynnistyessään päätyy salasananakyselyyn, jossa se syötettä odottaessaan ei käytännössä tee mitään.

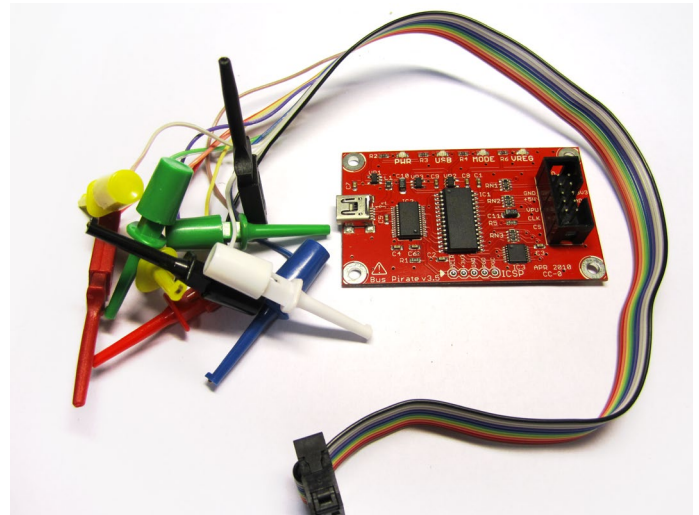
Thinkpad X61s -kannettavan purkaminen on hyvin suoraviivaista: avataan pohjasta ruuvit, jotka on merkitty pitämään näppäimistöä ja koteloa koossa. Tämän jälkeen näytön voi kääntää pitkälle auki (suoraksi asti) ja rungon kannen voi nostaa pois. Meitä kiinnostava komponentti löytyy Bluetooth-modulin alta. Kuvassa 4 on näkymä avatusta Thinkpad X61s -kannettavasta, josta Bluetooth-moduli on irrotettu ja sen paikka korostettu.

Suojamuovin voi kääriä sivuun, ja sen alta löytyy kuvan 5 mukainen osa emolevyä. Kuvassa näkyy korostettuna I²C-muistipiiri. Sen vierestä löytyvät näppärästi kontaktit, joihin mittapää voi kytkeä. Ammattilaiset voisivat kytkeä SMD-pihdit suoraan piirin jalkoihin, mutta pienen budjetin säätäjille riittää BusPiraten mittakoukkuun ripustettu nuppineula, jonka voi painaa kontaktiin. Jos kokeilet tätä ensimmäistä kertaa, suosittelemme pyytämään kaverin apuun lisäkäsiksi. Kolmas vaihtoehto on juottaa emolevylle lyhyet johdot kontaktipisteisiin ja kytkeä mittakoukut näihin johtoihin. Muista kytkeä myös BusPiraten maalinja (GND). Helpoin paikka tälle on vaikkapa Cardbus-korttipaikan suojaritilä.

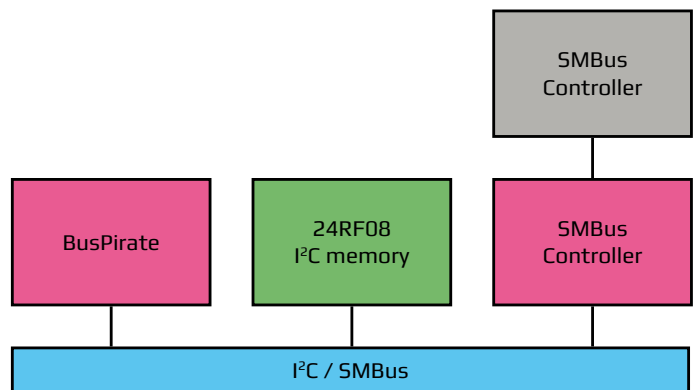
Kun kytkentä on selvä, voidaan siirtyä seuraavaan vaiheeseen. Tarkastellaan ensin, mitä tietokone väylällä juttelee. Tämän voi tehdä kytkemällä BusPirate ensin oikeaan moodiin:

1. "m" — vaihda moodi
2. "4" — I²C-moodi
3. "4" — ~400kHz nopeus

Jotta saadaan maistiaisen siitä, millaista viestintää tietokoneen väylällä kulkee, kytketään BusPiraten I²C-sniffer -makro päälle komennolla "(2)". Snifferimakrosta pääsee näppäimen painalluksella ulos. Nyt kun koneen käynnistää, terminaalin ruudulle tulisi syöksähtää muutaman kilotavun verran merkkimuotoista kuvausta I²C-väylän viestinnästä.



Kuva 2: BusPirate — hakkerin yleistyökalu



Kuva 3: BusPirate-kytkentä järjestelmän SMBus-väylälle

Taulukko 1: Näyte BusPiraten poimimasta I²C-liikenteestä

```
[0xA8+0x00+][0xA9+0xA7+][0xA8+0x01+][0xA9+0x8B+][0xA8+0x02+][0xA9+0xA5+][0xA8+0x03+][0xA9+0x47+][0xA8+0x04+][0xA9+0x29+][0xA8+0x05+][0xA9+0x19+][0xA8+0x06+][0xA9+0x03+][0xA8+0x07+][0xA9+][0xA8+0x10+][0xA9+0x81+][0xA8+0x24+][0xA9+0x41+][0xA8+0x25+][0xA9+][0xA8+0x26+][0xA9+0xA7+][0xA8+0x27+][0xA9+0x6F+][0xB8+0x06+][0xB9+][0xAC+0x10+][0xAD+][0xAC+0x11+][0xAD+0x25+][0xAC+0x22+][0xAD+][0xAC+0x23+][0xAD+0x07+][0xAC+0x25+][0xAD+][0xAC+0x26+][0xAD+0x1F+][0xAC+0x34+][0xAD+][0xAC+0x35+][0xAD+0x17+][0xAC+0x3F+][0xAD+0x11+][0xAC+0x40+][0xAD+0x0B+][0xAC+0x44+][0xAD+][0xAC+0x45+][0xAD+0x0D+][0xAC+0x4A+][0xAD+0x61+][0xAC+0x4B+][0xAD+0x07+][0xAC+0x4D+][0xAD+0xF1+][0xAC+0x4E+][0xAD+0x35+][0xAC+0x67+][0xAD+][0xAC+0x68+][0xAD+0x09+][0xAC+0x6B+][0xAD+][0xAC+0x6C+][0xAD+0x11+][0xAC+0x73+][0xAD+0x01+][0xAC+0x74+][0xAD+0x01+][0xAC+0x6D+][0xAD+0x01+][0xAC+0x6E+][0xAD+0x2D+][0xAC+0x6F+][0xAD+][0xAC+0x70+][0xAD+][0xAC+0x71+][0xAD+][0xAC+0x72+][0xAD+0x9F+][0xAE+0x54+][0xAF+0x01+][0xAE+0x57+][0xAF+0x41+][0xAE+0x38+][0xAF+0x3F+][0xAE+0x39+][0xAF+0x4B+][0xAE+0x3A+][0xAF+0x27+][0xAE+0x3B+][0xAF+0x31+][0xAE+0x3C+][0xAF+0x4D+][0xAE+0x3D+][0xAF+0x4D+][0xAE+0x3E+][0xAF+0x2F+][0xAE+0x28+][0xAF+0x01+][0xAE+0x29+][0xAF+0x01+][0xAE+0x2A+][0xAF+0x01+][0xAE+0x2B+][0xAF+0x01+][0xAE+0x2C+][0xAF+0x01+][0xAE+0x2D+][0xAF+0x01+][0xAE+0x2E+][0xAF+0x01+][0xAE+0x19+][0xAF+0x01+][0xAE+0x30+][0xAF+0x01+][0xAE+0x31+][0xAF+0x01+][0xAE+0x32+][0xAF+0x01+][0xAE+0x33+][0xAF+0x01+][0xAE+0x34+][0xAF+0x01+][0xAE+0x35+][0xAF+0x01+][0xAE+0x58+][0xAF+0x01+][0xAE+0x59+][0xAF+0x01+][0xAE+0x5A+][0xAF+0x01+][0xAE+0x5B+][0xAF+0x01+][0xAE+0x5C+][0xAF+0x01+][0xAE+0x5D+][0xAF+0x01+][0xAE+0x5E+][0xAF+0x01+][0xAE+0x5F+][0xAF+0x01+][0xAE+0x60+][0xAF+0x01+][0xAE+0x61+][0xAF+0x01+]
```

I²C-viestinnässä lähetettävät viestit aloitetaan niin sanotulla start-tilalla, jonka BusPirate esittää avaavalla hakasululla ("I"). Vastaavasti viestit päätetään niin sanotulla end-tilalla, joka esitetään sulkevalla hakasululla ("J"). Siirretyt tavut esitetään heksamuodossa ja kuitataan "+" ja "-" merkeillä riippuen siitä, oliko tapahtuma hyväksytty (ACK) vai hylätty (NACK). Tällöin esimerkiksi yhden tavun lähettäminen väylän slave-laitteelle tapahtuisi väylän master-laitteen sekvenssillä "[osoite data]". Slave-laite vastaisi jokaiseen lähetettyyn tavuun positiivisella kuittauksella ("+"). Väylää vakoileva BusPirate näkisi "[osoite+data+]". Tässä tulkinnassa "osoite" on siis I²C-väylälaitteen

osoite. Yleisenä käytäntönä osoitteen alin bitti on varattu merkitsemään, onko kyseessä luku- vai kirjoitusoperaatio.

AT24C08-piirin muistikapasiteetti on 1024 tavua, joka jakautuu neljään 256-tavuiseen lohkokoon, joilla kullakin on oma laiteosoite. Piirin komennustapa on hieman outo, mutta I²C-piireille tyypillinen. Kuvassa 5 on esitetty menettely tavun lukemiseen muisti-osoitteesta. Kuvasta näkyy, mitä datapinnissä SDA siirretään missäkin vaiheessa, ja alla on esitetty, mitä väylän isäntä ja laite vuorostaan sanovat. Komennon aloituksessa laite kuittaa isännälle, ja laitteen palauttaessa dataa isännälle tämä vuorostaan kuittaa. Positiivinen kuittaus jatkaisi lukuoperaatiota seuraavaan tavuun. Negatiivinen kuittaus tarkoittaa, että lukuoperaatio loppuu.

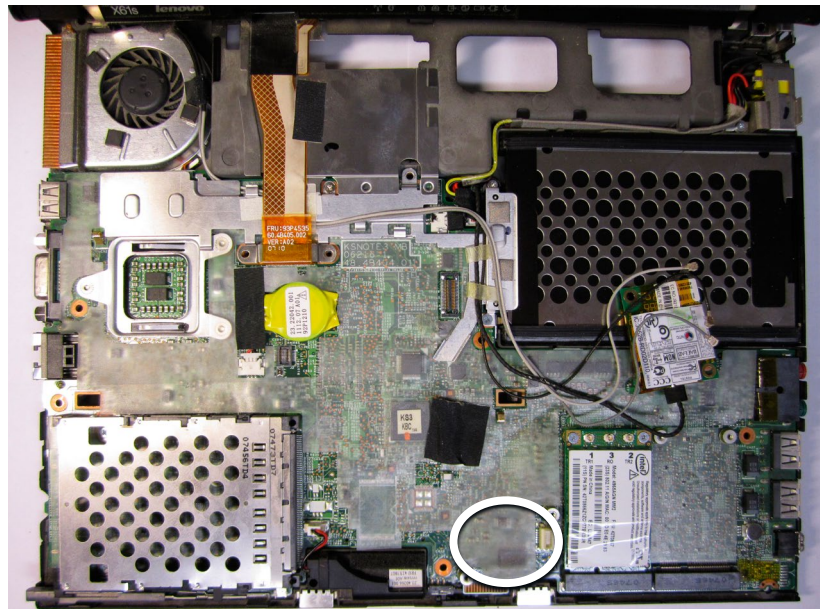
Väyläosoitteen ylin puolitavu on tällä piirillä aina 0xA (0b1010). Alempi puolitavu taas sisältää yhden sovelluskohtaisen kiinnitetyn bitin (Thinkpadin tapauksessa bitti on kiinnitetty arvoon 1) ja kaksi osoitusbittiä. Alin bitti määrittelee, onko kyseessä luku- vai kirjoitusoperaatio. Thinkpadin muistipiirin ensimmäinen lohko sijoittuu laiteosoitteille 0xA8 (kirjoitus) ja 0xA9 (luku). Muistiosoitteen toinen alempi tavu toimitetaan erikseen kirjoitusoperaation osana. Haluttaessa suorittaa lukuoperaatio satunnaiseen kohtaan muistia pitää siis aloittaa tekemällä kirjoitusoperaatio. Proseduuri olisi BusPirate-syntaksilla seuraava "[laiteosoite/kirjoitus alempioisotavun [laiteosoite/luku r]", missä "r" tarkoittaa tavun lukemista.

Thinkpadin tapauksessa ensimmäinen muistilohko sisältää laitteen sarjanumerot, toinen on varattu "käyttäjädatalle", kolmannessa on mallikoodit ja neljännessä suojausdata. Thinkpadien supervisor-salasana on melko usein tallennettu viimeisen lohkon muistiosoitteesta 0x38 alkaen näppäimistön scan-koodeina kaksi kertaa peräkkäin. Seuraavassa esimerkissä luumme 16 tavua tästä kohdasta muistia:

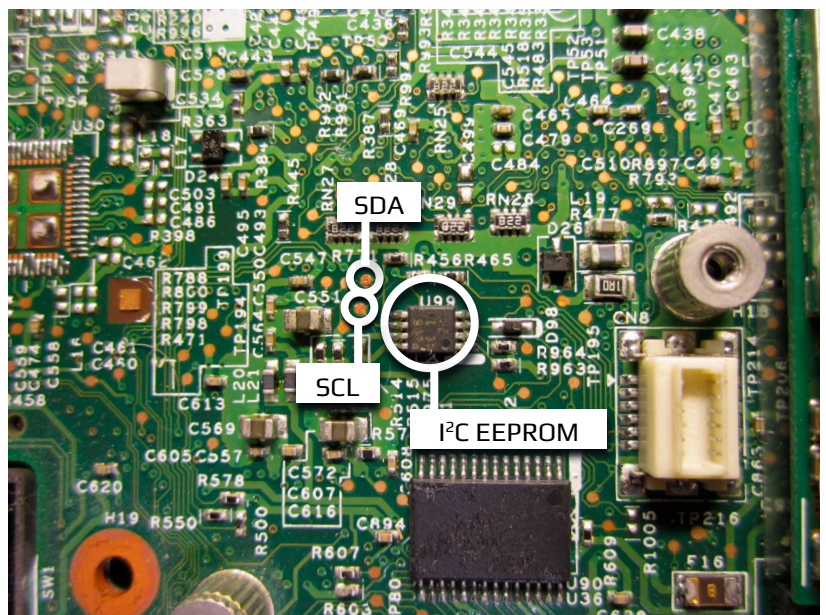
```
I2C>[0xAE 0x38 [0xAF r:16]
I2C START BIT
WRITE: 0xAE ACK
WRITE: 0x38 ACK
I2C START BIT
WRITE: 0xAF ACK
READ: 0x1F ACK 0x25 ACK 0x13 ACK 0x18 ACK 0x26 ACK 0x26
ACK 0x17 ACK 0xD2
ACK 0x1F ACK 0x25 ACK 0x13 ACK 0x18 ACK 0x26 ACK 0x26
ACK 0x17 ACK 0xD2

NACK
I2C STOP BIT
```

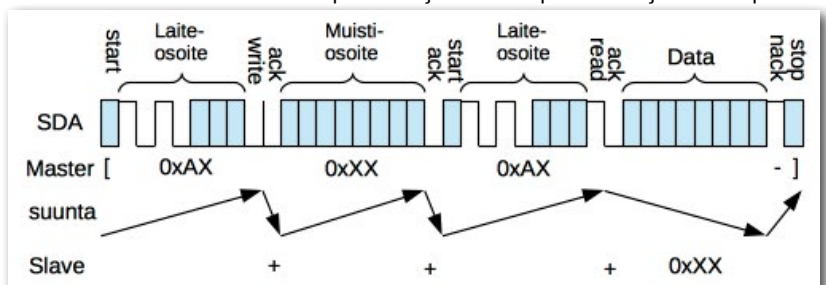
IBM SET 1 -näppäinkooditaulukon avulla voidaan tulkita, että koneen supervisor-salasana on "SKROLLI". Jos ei halua tulkita salasanaa, voi olla helpompaa ylikirjoittaa nykyinen salasana tällä tunnetulla. Sen voi tehdä komennolla "[0xAE 0x38 0x1F 0x13 0x18 0x26 0x26 0x17 0xD2 0x1F 0x25 0x13 0x18 0x26 0x26 0x17 0xD2]". Varsinainen poweron-salasana säilötään muualle, esimerkiksi TPM-piirille. Tämä tehdään myös jos käytössä on passphrase-asetus, jossa salasana on pidempi. Jos selkokielistä salasanaa ei näy, se saattaa olla tosiasiaa säilötty hinkpadin TPM-piirille. Tällöin keinona voi olla ylikirjoittaa salasana nollalla tai kopioida toisesta saman mallin suojaamat-



Kuva 4: Thinkpad X61s sisä näkymä



Kuva 5: Thinkpad X61s ja I²C-muistipiiri emolla ja kontaktipisteet



Kuva 6: AT24C08 I²C-lukuoperaatio. Lähde: AT24C08-datalehti

tomasta kannettavasta viimeinen 256 tavun lohko.

Viitteet:

- AT24C01A/02/04/08/16 Two-wire Automotive Serial EEPROM (3256F-SEEPR-10/04), Atmel, 2004, doc3256.pdf
- PC/AT Technical Reference, IBM, 1984.

SEURAAVAN SKROLLIN AIHEITA

Dwarf Fortress

OpenGL ja shaderit

Ohjelmoitavat Legot

CoolBasic -
skene

Hackerspacet

! Arduino

SQL-

injektiot

ja kuinka
suo jautua niiltä

\$\$

Saisinko
rakennettua
tietokoneen
keskiajalla

?

Algoritmi-
teoriaa

*
Symbolisen
tekoälyn
luomista

TUE SKROLLIA!

<http://skrolli.fi/tilaa>